



PHD

## Propagation Modelling for Urban Source Localization and Navigation

Dai, Zhuangzhuang

*Award date:*  
2019

*Awarding institution:*  
University of Bath

[Link to publication](#)

### Alternative formats

If you require this document in an alternative format, please contact:  
[openaccess@bath.ac.uk](mailto:openaccess@bath.ac.uk)

Copyright of this thesis rests with the author. Access is subject to the above licence, if given. If no licence is specified above, original content in this thesis is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC-ND 4.0) Licence (<https://creativecommons.org/licenses/by-nc-nd/4.0/>). Any third-party copyright material present remains the property of its respective owner(s) and is licensed under its existing terms.

#### Take down policy

If you consider content within Bath's Research Portal to be in breach of UK law, please contact: [openaccess@bath.ac.uk](mailto:openaccess@bath.ac.uk) with the details. Your claim will be investigated and, where appropriate, the item will be removed from public view as soon as possible.



*Citation for published version:*

Dai, Z 2018, 'Propagation Modelling for Urban Source Localization and Navigation', Ph.D., University of Bath.

*Publication date:*

2018

*Document Version*

Publisher's PDF, also known as Version of record

[Link to publication](#)

Attention is drawn to the fact that copyright of this thesis rests with its author. This copy of the thesis has been supplied on the condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the prior written consent of the author. This thesis may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

## University of Bath

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# Propagation Modelling for Urban Source Localization and Navigation

submitted by

Zhuangzhuang Dai

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Electrical and Electronic Engineering

September 2017

## **COPYRIGHT**

Attention is drawn to the fact that copyright of this thesis rests with its author. This copy of the thesis has been supplied on the condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the prior written consent of the author.

This thesis may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

Signature of Author .....

Zhuangzhuang Dai

# Acknowledgement

My sincere thanks to all the people who give me support during my pursuit for PhD.

My deepest gratitudes go to my supervisors, Dr Robert Watson and Dr Peter Shepherd. Especially Dr Watson, I appreciate his scientific passion, profound knowledge and experience which always guide me towards a right direction, and offering me an open minded research environment. His dedication and commitment to research keeps inspiring me during my study.

I am grateful the Electrical and Electronic Engineering department of University of Bath supports me financially, and the faculty and colleagues have been helpful and patient.

I want to express my thankfulness to my parents who love and care about me unconditionally.

I lovingly dedicate this thesis to my wife, Dr Xuan Wang.



# Abstract

This thesis investigates a propagation modelling approach to navigation and source localization. Complex urban propagation environments give rise to severe multipath which impairs the reliability of conventional satellite and terrestrial based localization systems. The motivation is the development of a location determination scheme exploiting multipath propagation. In this thesis a new ray tracing method has been developed to efficiently determine channel characteristics. Using a database of channel characteristics, a model is constructed to determine location of a receiver based on a matching algorithm. This thesis also investigates into the inverse problem, i.e., the source localization in urban environments. These methods have been tested against noise and perturbations and is shown to be robust. In the simulated urban environments, navigation errors are typically less than 15 m. The proposed source localization algorithm is able to locate a radio source to better than 100 m.

To evaluate these algorithms a 2.5D ray launching model has been developed, which is able to make use of existing digital map databases. The accuracy of this model has been validated by comparing simulated Received Signal Strengths (RSS) against channel sounding measurements in the city center of Munich, Germany. The ray launching model makes use of parallel computing techniques using Graphic Processing Units (GPU).

Key to the success of these methods is a new fingerprinting technique which correlates abstract electromagnetic features with physical coordinates. This takes advantage of data mining and machine learning to study patterns in the fingerprint distribution. The thesis details the implementation of the navigation and source localization algorithms. In particular, consideration is given to the problem of source localization using a small Unmanned Aerial Vehicle (UAV). In order to efficiently solve this problem, the technique of Dynamic Time Warping (DTW) has been explored. All errors have been quantified and their sensitivities are determined.

# Contents

<b>Acknowledgement</b> . . . . .	i
<b>Abstract</b> . . . . .	ii
<b>List of Figures</b> . . . . .	vi
<b>List of Tables</b> . . . . .	ix
<b>List of Abbreviations</b> . . . . .	1
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	3
1.2 Localization system decomposition . . . . .	5
<b>2 Literature review</b>	<b>9</b>
2.1 Traditional localization and signals of opportunity . . . . .	9
2.2 Propagation modelling . . . . .	12
2.3 Digital mapping . . . . .	18
2.4 Fingerprinting . . . . .	19
2.5 Artificial neural network . . . . .	20
2.6 Pattern recognition . . . . .	21
2.7 Parallel computing . . . . .	23
2.8 Summary . . . . .	25
<b>3 Ray launching model</b>	<b>26</b>
3.1 3D modelling . . . . .	27
3.2 Environmental entries . . . . .	28
3.2.1 Antennas and propagation media . . . . .	28
3.2.2 Urban layout entry . . . . .	29
3.2.3 Trees . . . . .	32
3.3 Propagation primitives . . . . .	33
3.3.1 Reflection . . . . .	34
3.3.2 Diffraction . . . . .	36

3.3.3	Wall penetration . . . . .	38
3.3.4	Roof-top diffraction . . . . .	39
3.4	Field reconstruction . . . . .	43
3.5	2.5D ray launching model validation . . . . .	47
3.6	Summary . . . . .	52
<b>4</b>	<b>Navigation</b>	<b>53</b>
4.1	Location fingerprint . . . . .	53
4.2	Localization algorithm . . . . .	59
4.2.1	Mapping function . . . . .	59
4.2.2	Training . . . . .	61
4.3	Accuracy and reliability . . . . .	68
4.3.1	Number of opportunistic sources . . . . .	69
4.3.2	Location fingerprint components . . . . .	69
4.3.3	Location target resolution . . . . .	70
4.3.4	ANN configuration . . . . .	71
4.3.5	Noise level during measurement . . . . .	73
4.3.6	Environmental factors . . . . .	74
4.4	Summary . . . . .	77
<b>5</b>	<b>Source localization</b>	<b>78</b>
5.1	Source localization algorithm . . . . .	78
5.1.1	Data collector . . . . .	79
5.1.2	Feature transformation . . . . .	81
5.1.3	Source location fingerprint . . . . .	82
5.1.4	Matching algorithm . . . . .	83
5.2	Accuracy of analysis . . . . .	86
5.2.1	Measurement error and path deviation . . . . .	87
5.2.2	Noise level and bias . . . . .	92
5.2.3	Route selection . . . . .	97
5.3	Summary . . . . .	106
<b>6</b>	<b>Enhancement and program optimization</b>	<b>107</b>
6.1	Sequence-based navigation scheme . . . . .	108
6.2	Matlab . . . . .	109
6.3	GPU-based acceleration . . . . .	112
6.4	Summary . . . . .	118

<b>7 Summary and future work</b>	<b>119</b>
7.1 Summary . . . . .	119
7.2 Future work . . . . .	120
<b>References</b> . . . . .	<b>122</b>
<b>Appendix</b> . . . . .	<b>137</b>

# List of Figures

1-1	Flowchart of navigation system . . . . .	6
1-2	Flowchart of source localization system . . . . .	7
2-1	Schematic diagram of satellite navigation . . . . .	10
2-2	Ray tube . . . . .	14
2-3	Bounding box division . . . . .	15
2-4	Google Earth . . . . .	18
2-5	NVIDIA GTX 280M . . . . .	24
3-1	3D ray launching model . . . . .	27
3-2	Radiation pattern . . . . .	29
3-3	Overpass Turbo panel . . . . .	30
3-4	Propagation paths at the Arc de Triomphe . . . . .	31
3-5	Transmission through a tree . . . . .	33
3-6	Reflection geometry . . . . .	35
3-7	Diffraction geometry on edge . . . . .	36
3-8	Transmission geometry . . . . .	38
3-9	NYC street canyon . . . . .	40
3-10	Roof top diffraction geometry . . . . .	41
3-11	Reception detection mechanism . . . . .	43
3-12	Enclosed box scenario with seven objects inside . . . . .	45
3-13	Power delay profile . . . . .	46
3-14	Angular Spread . . . . .	46
3-15	Measurement campaigns in Munich . . . . .	47
3-16	Pathloss measurements on Munich scenario . . . . .	48
3-17	Simulated RSSs along ‘METRO 200’ . . . . .	49
3-18	Simulated RSSs along ‘METRO 201’ . . . . .	50
3-19	Simulated RSSs along ‘METRO 202’ . . . . .	50
3-20	Scatter plot between simulated and measured pathloss along ‘METRO202’	52

4-1	Propagation paths in virtual scenario . . . . .	55
4-2	PDP at NLOS receiver . . . . .	56
4-3	Angular Spread at NLOS receiver . . . . .	56
4-4	Fingerprints generated by opportunistic sources . . . . .	58
4-5	RSS heat map of a virtual scenario . . . . .	60
4-6	Artificial Neural Network . . . . .	61
4-7	Hoxton district layout . . . . .	62
4-8	RSS heat map . . . . .	63
4-9	AOA x-axis Cartesian projection distribution . . . . .	64
4-10	AOA y-axis Cartesian projection distribution . . . . .	65
4-11	Neural Network user interface . . . . .	66
4-12	Neural Network settings . . . . .	66
4-13	Neural Network configuration . . . . .	67
4-14	Training result . . . . .	68
4-15	Mean error versus number of sources . . . . .	69
4-16	Mean error versus target resolution . . . . .	71
4-17	Localization accuracy versus noise strengths . . . . .	73
4-18	Localization accuracy versus measurement biases . . . . .	74
4-19	Tests <i>A</i> and <i>B</i> illustration . . . . .	75
4-20	Tests <i>C</i> bus insertion . . . . .	76
5-1	Static vs. dynamic remote sensing . . . . .	79
5-2	Dat collector: UAV . . . . .	80
5-3	Anticipated waveforms . . . . .	82
5-4	Simulated UAV route . . . . .	83
5-5	Dynamic Time Warping versus Euclidean matching . . . . .	85
5-6	Route deviation cases (a) and (b) on Hoxton scenario . . . . .	88
5-7	Route deviation cases (c) and (d) on Hoxton scenario . . . . .	89
5-8	Fingerprint characteristics of ‘METRO 200’ . . . . .	91
5-9	Normalized accuracies versus noise strengths . . . . .	92
5-10	Normalized accuracies versus biases . . . . .	93
5-11	Mean error distance versus noise strengths . . . . .	93
5-12	Mean error distance versus biases . . . . .	94
5-13	Accuracy versus biases after swapping coordinates . . . . .	95
5-14	Mean error distance versus biases after swapping coordinates . . . . .	96
5-15	Routes of measurement in Munich . . . . .	97
5-16	Pathloss Measurements on Munich scenario . . . . .	98

5-17	RSS heat map on Munich scenario . . . . .	99
5-18	Measurement routes against RSS heat map of Munich scenario . . . . .	100
5-19	Simulated RSSs along ‘METRO 200’ . . . . .	101
5-20	Simulated RSSs along ‘METRO 201’ . . . . .	101
5-21	Simulated RSSs along ‘METRO 202’ . . . . .	102
5-22	Localization error versus noise strengths and bias . . . . .	103
5-23	Fingerprint of ‘METRO 200’ . . . . .	104
5-24	Fingerprint of ‘METRO 201’ . . . . .	105
5-25	Fingerprint of ‘METRO 202’ . . . . .	105
6-1	Sequence-based navigation method . . . . .	108
6-2	A flowchart of master program . . . . .	110
6-3	Matlab profiling report . . . . .	112
6-4	GPU memories and multiprocessors . . . . .	113
6-5	Computation time versus CPU/GPU and scenario complexity . . . . .	115
6-6	Visual Profiler results . . . . .	116
6-7	Visual Profiler zoomed-in on one iteration . . . . .	116
6-8	GPU usage analysis . . . . .	117

# List of Tables

3.1	Tree loss model parameters . . . . .	32
3.2	Model validation against Munich measurement campaign . . . . .	51
4.1	Emitter allocations . . . . .	62
4.2	Transfer functions . . . . .	67
4.3	Mean errors against fingerprint primitives . . . . .	70
4.4	Accuracy versus number of neurons in each layer . . . . .	72
4.5	Accuracy versus training gain . . . . .	72
4.6	Training specifications and performances on different scenarios . . . . .	75
4.7	Moving bus interference test . . . . .	77
5.1	Source localization case studies . . . . .	86
5.2	Accuracy performance versus deviations and measurement errors . . . . .	87
5.3	Accuracy against Width of adjustment window . . . . .	96
6.1	Computation time analysis . . . . .	114



# List of Abbreviations

<b>ANN</b>	Artificial Neural Network
<b>AOA</b>	Angle of Arrival
<b>AWGN</b>	Additive White Gaussian Noise
<b>BER</b>	Bit Error Rater
<b>CDF</b>	Cumulative Distribution Function
<b>CPU</b>	Central Processing Unit
<b>CUDA</b>	Compute Unified Device Architecture
<b>DFT</b>	Discrete Fourier Transform
<b>DTW</b>	Dynamic Time Warping
<b>FDOA</b>	Frequency Difference of Arrival
<b>FEM</b>	Finite Element Modelling
<b>GNSS</b>	Global Navigation Satellite System
<b>GPS</b>	Global Positioning System
<b>GPU</b>	Graphics processing units
<b>GTD</b>	Geometric Theory of Diffraction
<b>IoT</b>	Internet of Things
<b>LOS</b>	Line of sight
<b>LPF</b>	Low-pass filter
<b>MED</b>	Mean Excess Delay
<b>MoM</b>	Method of Moments
<b>MSE</b>	Mean Square Error
<b>NLOS</b>	Non-line of sight
<b>PDP</b>	Power Delay Profile
<b>RF</b>	Radio Frequency
<b>RFID</b>	Radio Frequency Identification Devices
<b>RMSDS</b>	Root-mean-square delay spread
<b>RSS</b>	Received Signal Strength
<b>SHF</b>	Super High Frequencies
<b>SNR</b>	Signal to Noise Ratio
<b>TDOA</b>	Time Difference of Arrival
<b>UAV</b>	Unmanned Aerial Vehicle
<b>UTD</b>	Universal Theory of Diffraction
<b>WGN</b>	White Gaussian Noise

# Chapter 1

## Introduction

Wireless communication has flourished in the past decades. With growing demands on mobile telecommunication and internet connectivity, wireless communication links are multiplying on the ground, in the air and on the sea. Since lower radio frequencies, below 2 GHz, are becoming crowded, capacious bandwidths at higher frequencies are expected to support downloading, video and audio streaming with improved data rates. Empirical path loss models are rarely completely satisfactory in multipath environments, especially in urban areas where buildings and structures introduce complicated propagation behaviour. On the other hand, this multipath interference directly results from the propagation environment which can be exploited to provide geographical information through appropriate interpretation of the channel characteristics.

Traditional localization methods, e.g., satellite navigation systems, are mostly based on a line-of-sight (LOS) propagation assumption. In this thesis, we are inspired to mitigate blind spots of traditional navigation by creating an urban positioning approach. It is well recognized that deterministic methods for wave propagation modelling are among the best methods for the estimation of radio channel characteristics. With the aid of environmental specifications (e.g., building position and materials), deterministic propagation models, such as ray tracing, imaging theory and Method of Moments (MoM), can provide attractive channel response simulations that compare well to measurement. Such models could guide off-line solutions to coverage prediction, radio planning, navigation and source localization.

There is current interest in the determination of position and navigation in situations where satellite navigation signals from GNSS (Global Navigation Satellite System) are denied. The denial of the GNSS signals may be via some form of intentional signal jamming, unintentional through interference or merely a consequence of the challenging environment, e.g., tall buildings in ‘urban canyons’. In these circumstances knowledge

of the propagation environment can help localize the source of the interference or alternatively be used in conjunction with signals of opportunity to aid a navigation solution.

Preliminary work of this thesis focuses on the development of a wave propagation model suitable for urban environments. Then we aim at establishing mobile and source localization algorithms through machine learning of the channel characteristics with respect to location.

A 2.5D ray launching model was designed, implemented and tested by the author, which is shown to be adequately efficient and accurate for the development of navigation and source localization algorithms. It utilizes digital map databases, which contain building and tree coordinates, in conjunction with semi-deterministic propagating engines, including specular reflection, vertical edge diffraction, wall penetration and roof-top diffraction. Power Delay Profile (PDP) and Angular Spread are the deliverables from the reception capture.

Feature eigenvectors, also termed ‘fingerprints’, can be extracted from the channel modelling results to assist training and learning of their geographical distribution pattern. This leads to a mobile localization algorithm via Artificial Neural Network (ANN) which are powerful in dealing with discontinuous high-dimensional data sets such as these. The main objectives of this thesis are the generalization of such learning processes on different urban scenarios as well as validation of its stability and reliability.

Source localization is an ill-posed inverse problem which involves a more careful definition of fingerprint. We propose to use an Unmanned Aerial Vehicle (UAV) as a probe to capture variations of trends of electromagnetic parameters along a specified path. The outstanding frequency components are taken as fingerprints to check against simulated source location candidates. A distance measuring algorithm, called Dynamic Time Warping (DTW), is adapted to determine the best matched source location. This method is shown to be accurate and robust.

This thesis aims to establish a reliable navigation and source localization system based on propagation modelling in conjunction with fingerprinting and pattern recognition techniques. The accomplishment of this objective requires exploration of the questions below:

- What difficulties confront current navigation systems in urban environments?
- Is there any reliable and efficient radio source localization method in urban scenarios?
- What is the best approach to construct a navigation and source localization system?

- What are the key requirements in order to exploit channel characteristics to aid a positioning solution?
- What is the accuracy performance of a deterministic wave propagation model in urban environments?
- What methods can be used to enhance the speed of deterministic wave propagation models?
- What are the accuracy and reliability of such navigation and source localization methods?

In order to address these questions, this thesis is organized as follows (see also Figures 1-1 and 1-2). The remainder of this chapter expands the underlying foundation and motivation for this research. Chapter 2 reviews previous technologies and researches existing urban localization methods, and explains their strengths and weaknesses. Various wave propagation models are investigated and a decision is made that ray launching models are sufficiently reliable in generating channel characteristics. Unrestricted access to digital maps is also demonstrated via the OpenStreetMap project. Current techniques of data mining and machine learning are investigated which form the basics of fingerprint extraction. This chapter also reviews multi-thread acceleration methods for ray tracing.

Chapter 3 presents the development of our 2.5D ray launching model in detail, containing antenna simulation, map abstraction, propagation mechanisms, reception detection and validation. Chapter 4 explains the definition of location fingerprint and how we utilize ANN to generate a mobile localization algorithm. The neural network obtained exhibits convincing reliability and accuracy against noise and data corruptions. Chapter 5 demonstrates the definition of source location fingerprint with the aid of UAV measurements, and also the development of a matching algorithm. The stability of the source localization method is tested against a series of perturbations, such as route deviation, wind conditions, background noise strength and bias etc. where satisfactory results are found. Acceleration and optimization methods are discussed in Chapter 6. Finally, a summary and future work are presented in Chapter 7.

## 1.1 Background

Modern life is heavily reliant upon satellite navigation. From a military defence system to daily driving, GNSS systems, such as the Global Positioning System (GPS), are playing an important part. However, the propagation between satellite and ground

terminal can be vulnerable to multipath, bad weather or intentional GPS jamming. Multipath propagation, especially in densely built areas, undermine the stability of satellite navigation due to near ground reflections. Unpredictable weather conditions and atmospheric properties bring about complex refraction and scattering into the air-to-ground propagation. Alternative user localization plans are needed in such circumstances to make up for the weakness of GPS.

Jamming devices are radio frequency (RF) transmitters that intentionally block, jam, or interfere with lawful communications, such as cell phone calls, text messages, GPS systems and Wi-Fi networks [123]. Such jammers are illegal to use but unfortunately very easy to buy and cheap to build. The consequence of GPS jamming can be severe, e.g., allowing suspects to hide criminal activities. In this case, being able to locate the jamming transmitter becomes a focus for security. Source localization in multipath environments has been a long-standing problem, because triangulation calculations cannot be directly used on corrupted non-line of sight channels. Previous researchers have investigated whether multipath information can be exploited to aid a positioning solution. Results testify the possibility as long as channel response is clearly interpreted and a reliable localization algorithm is drawn.

The wireless communication technique has seen significant development in the past decades, which brings great interest in comprehending the wave propagation characteristics. As a result, channel response, radio location, as well as base-station planning are all resolvable through effective propagation modelling instead of laborious measurements.

Propagation modelling has been constantly evolving from statistical modelling of propagation distance to deterministic 3D modelling for specific environments. This technique is becoming mature in both efficiency and reliability. With detailed specifications of the propagation environment, there is growing reason to trust wave propagation models to take the place of measurement campaigns which are both costly and subject to errors due to temporal changes.

Data mining and machine learning, including the recent hot-spot ANN, enable intelligent, adaptive and rapid generation of classification or regression functions. In particular, pattern recognition could be applied to linking channel characteristics with locations to formulate a localization algorithm. The off-line processing relies on a large collection of training resources which can be extracted from the channel modelling. This feature extraction based on physical characteristics, works in conjunction with fingerprinting, a numerical transformation which quantifies abstract notions as vectors or a matrix. A successful definition of fingerprint and an appropriate mapping function would make a robust real-time localization approach.

The ever-growing computational capacity of digital processors, the wide-spread use of mobile transceivers and easy access to digital city maps altogether create a solid foundation for the formulation of the localization procedure. In terms of hardware, cell phones, sensor networks as well as mobile carriers are able to fulfil cheap data collection. State-of-the-art multi-core processors guarantee an efficient data analysis power, even on embedded mobile devices. From a software perspective, a suitable channel modelling tool, which is currently a relatively mature technique, and digital map databases offer comprehensive urban simulation capability which can generate a large amount of electromagnetic characteristics data.

These observations and the encouraging initial results of previous researchers, as well as many industries inspire the investigation of this novel urban localization approach, which overcomes drawbacks of traditional navigation systems particularly in multipath environments. From an engineering point of view, the required advanced hardware facilities and software platforms are almost in place and the cost of such systems continues to reduce.

## 1.2 Localization system decomposition

Through a comprehensive review of the literature in Chapter 2, we conclude that deterministic wave propagation modelling continues to be a state-of-the-art channel estimation technique, thanks to the increasing power of modern processors and digitization of map databases. The extensive use of wireless communication links provides valuable information on channel characteristics in the context of Internet-of-Things (IoT) and Big Data. We are, thus, exploring the possibility of digging and mining this data from an information perspective. A ray launching model is developed to generate such channel characteristic data, which is demonstrated in Chapter 3, to take the place of laborious measurement. Based on our research of previous measurements and validations, we believe 2.5D ray launching models are sufficiently accurate and efficient in urban environments. Acceleration and optimization of the ray launching program is detailed in Chapter 6.

In this research we propose to investigate the feasibility of a model aided navigation and source localisation approach. The basic hypothesis is that for a given environment and a given source location, the trends in key signal characteristics such as Received Signal Strength (RSS), Time of Arrival (TOA) and Angle of Arrival (AOA) as determined at various points throughout the environment, will be sufficient to determine its location. This hypothesis happens to share a common idea with the ‘fingerprinting’ technique which identifies a target by extracting distinctive features and exhaustively

searching for the best match.

Therefore, we implemented a navigation system based on the extraction and recognition of ‘location fingerprints’. As shown in Figure 1-1, the fingerprint database is built from propagation modelling results with reference to the environmental entries (building location and materials). Given the radio source and a grid of reference locations, the 2.5D ray launching model is used to generate ‘truth data’ for these locations. Then, we adopt an ANN to help construct a deterministic mapping function between the location fingerprints and geographical locations. Integration and evaluation of this navigation process is expanded in detail in Chapter 4. A sequence-based navigation scheme based on the fingerprint algorithm is introduced in Chapter 6 to improve the off-line utility of the navigation system.

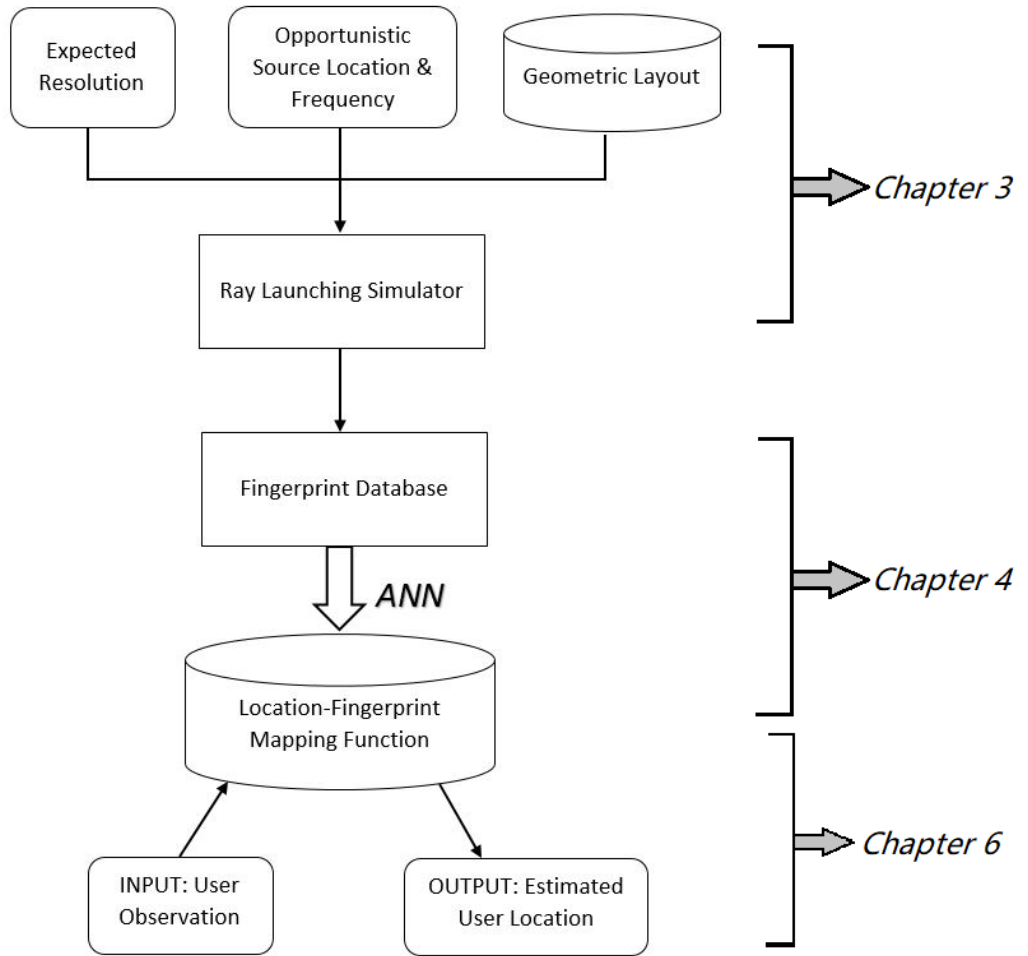


Figure 1-1: Flowchart of the navigation system with respect to content location.

If one considers a ‘Manhattan-style’ city layout, it is clearly possible to determine

the location of a source in an arbitrary location if one traverses all possible streets and selects the largest RSS. However what is not clear is if one can derive generalised rules for selecting the shortest trajectory of say a receiver payload aboard a small UAV to achieve acceptable accuracy. This is a novel combination of both an inverse problem and a propagation problem.

The core of a source localization system is to estimate, *a-priori* for a given known environment, the minimum trajectory of receiver measurement locations in order to best locate the source. This also encourages the discrimination of ‘source location fingerprints’ as long as the UAV observations differ from each source location to another.

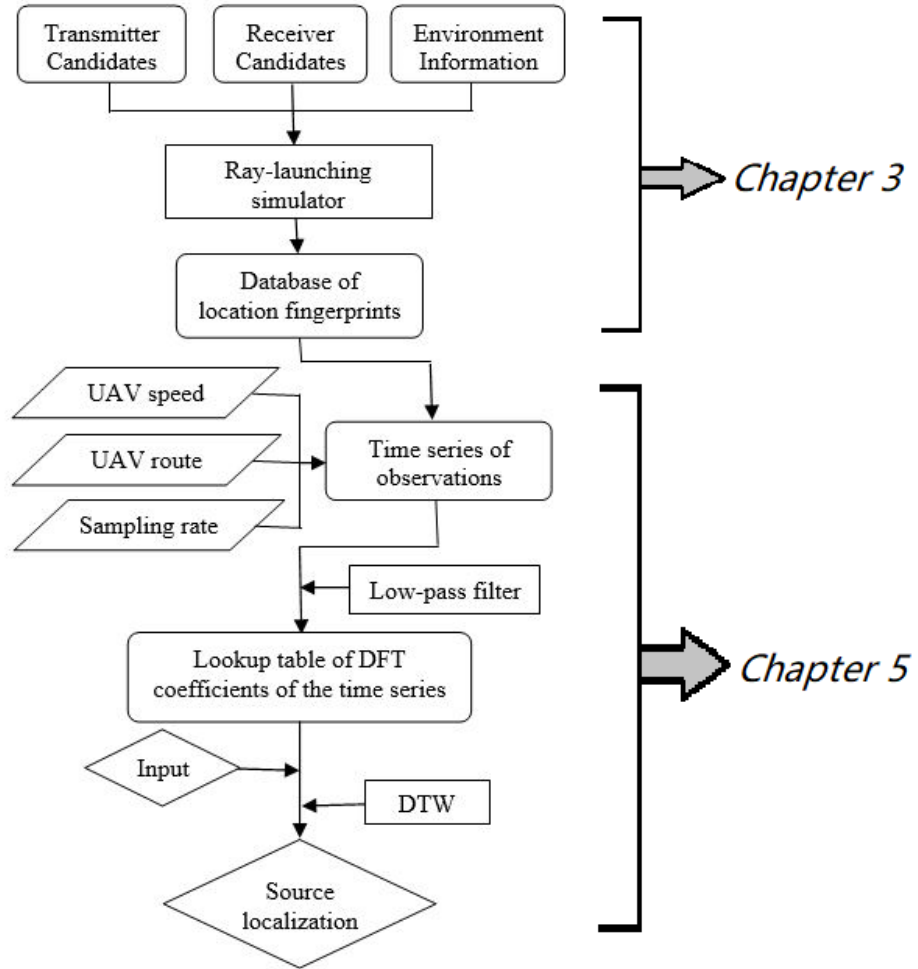


Figure 1-2: Flowchart of the source localization system with respect to content location.

As a result, we are inspired to define a ‘source location fingerprint’ based on UAV data collection along a specified path in the street canyons. Procedures of this source localization approach are shown in Figure 1-2. In this, the ray launching model plays



a significant role in generating UAV observations along the path with respect to source reference locations. As the number of source reference locations increases, the potential precision of localization increases, whereas, the price is a much higher computational cost. From this data pattern recognition techniques are adopted to estimate the source location as detailed in Chapter 5. The Discrete Fourier Transform (DFT) and Dynamic Time Warping (DTW) algorithms are combined to implement the fingerprint matching algorithm. The source localization performance as a function of various errors and perturbations, as well as route selection is further investigated in Chapter 5.

In comparison to GPS (Global Positioning System) techniques the proposed mobile and radio source localization approaches can employ lower frequencies (e.g., VHF/UHF for DAB/DVB compared to L-band for GNSS), the use of which greatly reduces the effects of building blockage while purposefully exploiting multipath propagation. A key advantage over conventional fingerprinting methods is the ability to generate navigation information without the need to make detailed and costly measurements of the environment.

To obtain fingerprints from modelling is cost effective and more flexible. If the environment is changed due to, e.g., building construction or destruction, a set of fingerprint database can be regenerated within hours using our acceleration techniques as is illustrated in Chapter 6. Moreover, case studies and test results confirm that our localization approaches possess a considerable degree of robustness in the presence of noise and interference. This suggests that an absolutely precise environmental description is not necessary.

Note that in this work we assume a radio source is always static in terms of time and position during measurement. Within the area of interest, it is regarded as the only transmitter such that there is no cross-interference with other emitters. These opportunistic signals are deemed to be narrow-band such that frequency dispersion is not a concern. Last but not least, we assume that the simple devices pervasively equipped on user mobiles and the UAV, are only able to measure values of RSS, TOA and AOA with limited accuracy.

## Chapter 2

# Literature review

This chapter provides a comprehensive review of propagation and localization related literature. Conventional satellite based and terrestrial localization methods are both found to be less effective in densely built areas mostly due to multipath propagation. In recent decades wireless communication links have seen remarkable growth in distribution density as well as frequency range. This allows the possibility to use signals of opportunities received by mobile devices which could be exploited for localization use. To do this, it is crucial to establish a channel characteristics database in terms of geographical location before an on-line localization method can be established. Categorization and features of different propagation models are investigated. It is concluded that a 2.5D ray launching model is a reliable and efficient tool for urban scenarios which are accessible from rich digital map resources. Fingerprint-based data mining and machine learning techniques are reviewed in pursuit of a reliable function bridging the channel characteristics with physical locations. State-of-the-art parallel computing methods using general purpose graphic cards have been explored to investigate the capability of executing rapidly the propagation modelling stage.

### 2.1 Traditional localization and signals of opportunity

Currently, satellite navigation systems are the most widely used localization approach [49]. A built-in electronic receiver for the satellite constellation system is necessary to achieve autonomous geo-spatial positioning. The local time of the receiver can also be determined with high precision to accomplish synchronization. Global Navigation Satellite System (GNSS) refers to any satellite navigation system with global coverage. Up to now, mature GNSSs include the European Union's Galileo system, the Russian GLONASS, China's BeiDou system and the United States NAVSTAR

Global Positioning System (GPS). The GPS system provides the broadest application for military, civil and commercial use. The GPS system does not require the user to transmit any data, and operates independently of any telephonic or internet reception [31], although the usefulness of the GPS positioning can be improved by these assistant functions. A schematic diagram of satellite navigation application is shown in Figure 2-1.

The accuracy of GNSS localization is typically very high, usually less than 5 m [16]. Nevertheless, line-of-sight (LOS) propagation between satellite and receiver is critical because theoretically time delays from at least three satellite signals are required to allow positioning to an exact point. However, this hypothesis does not always hold in densely built urban areas [125]. Flat building surfaces typically act as reflectors creating receiver images which undermine the satellite localization precision [29]. Severe multipath propagation in urban environments also makes synchronization problematic. Moreover, the satellite-to-ground propagation path may be distorted merely as a consequence of bad weather.



Figure 2-1: Schematic diagram of satellite navigation on ground, in ocean and air, taken from: <https://www.laboratoryequipment.com/news/2013/03/ground-system-improves-satellite-navigation>.

Terrestrial communication links can also be used for geo-positioning. Such methods rely on a trilateral geometric estimation which requires three mobile link measurements

of TOAs or AOAs. Such terrestrial localization is also affected by multipath from high-rise typical of modern cities. Furthermore, TOA and AOA observations may not be due to LOS propagation as expected.

Most traditional localization methods lack reliability in urban localization scenarios which has seen growing interest in recent decades. Many attempts have been made to exploit knowledge of the propagation environment in conjunction with signals of opportunities to aid the localization solution. Schmidt [108] made exploratory research into the diversity of signal parameters which can be used to locate radio sources. The usability of RSS, TOA and AOA in terms of indoor localization are discussed in [14, 19, 24] where good accuracies are found through experiments. In [63, 83], the authors developed an efficient ray tracing tool for indoor localization. However, these models are too simplistic to see wide application for complex urban scenarios.

Huang *et al.* [45] proposed the use of weighted averaged RSSs of wireless communication links for indoor navigation. This approach presented very good reliability in the presence of measurement noise. Schmitz *et al.* [110] tested the feasibility of using TDOA (Time Difference of Arrival) as an indicator to localize mobile users in urban environments. The merit of this approach is that it purposefully takes advantage of the multipath propagation. However, reliability tests when subject to noise and interferences are not given. Del Corte *et al.* [22] put forward a source localization approach which only requires several AOA measurements without any knowledge of the environment. A key requirement is the determination of the dominant ray and its delay, which is problematic. Kelner *et al.* [55] attempted to locate the emitter based on the signal Doppler frequency. However, little validation for different environments is given.

All of the previously described localization methods have been developed using signals of opportunity. A signal of opportunities can be considered to be a signal transmitted for non-navigation purposes but from which time of flight and other useful information can be determined [52, 87, 103]. In urban environments, the growing number of wireless communication links, e.g., Wi-Fi sources, offer rich resources of opportunistic signals with spatial diversity. Collecting and comprehending these signals could not only help localize mobile users but reveal unknown source locations through an inverse approach.

The study of the complete propagation channel including antenna radiation pattern is the key to successfully comprehending opportunistic signals. Knowledge of the antenna pattern is well known for the types of antenna typically used in mobile user equipment. Taking mobile phone for example, the antennas range from a planar inverted F antennas, a planar meander line, a folded loop to a modified dipole. The major part of channel reconstruction relies on multipath propagation modelling which

will be discussed in the next section.

## 2.2 Propagation modelling

The physical layer properties of communication systems have drawn increasing scientific and industrial interest in the past few decades. The lower frequency bands of the microwave range have been well studied and pervasively used [102]. A major advantage of using lower frequency signals is that free space propagation loss and building penetration losses are reduced [84]. Thus, it is feasible to broadcast over large distances with limited transmitted power.

Traditional radio planning methods typically rely heavily on expensive as well as time consuming field measurements. From these measurements, empirical wave propagation models, such as Okumura-Hata, COST-231, Walfish-Bertoni models have been developed to provide rapid path loss estimation in typical urban environments [12, 101]. However, many of these models fail when applied to other frequencies and scenarios. Furthermore, measurement-based calibration may ultimately be required.

Since plenty of applications are saturating the lower RF bands, modern wireless communications are seeking wider bandwidths, so as to increase data rate by migrating to the higher microwave frequency band [25]. For example, 2.45 GHz ISM bands, 6 GHz and up to 60 GHz bands are being investigated for 5G applications [42, 43, 102]. However, these signals suffer higher propagation loss and encounter more complicated scattering phenomena [139]. Deterministic models are thus being developed. Full-wave models, e.g., Method of Moment (MoM), can solve the wave propagation behaviour based on Maxwell's equations [48]. Although they provide good accuracy, the computational cost rises exponentially as the size of the environment increases.

Ray tracing and ray launching models based on geometrical approximations can also be categorized as deterministic models [97]. They are state-of-the-art techniques for indoor modelling thanks to the comprehensive inclusion of environmental details which essentially account for multipath propagation. Both ray tracing and ray launching methods use the concept of ray optics but employ different approaches.

Ray tracing is more of an objective-oriented simulation. It constructs images of virtual sources or receivers with respect to object surfaces to identify potential path. A visibility tree is often adopted to establish the network of consecutive reflections and diffractions [7, 135]. A representative application of ray tracing is the corridor model in which different orders of reflections bouncing along a tunnel are assumed to be the potential propagation paths [15]. Corridor models are very effective and efficient in indoor modelling and may be applied to urban street canyons [6].

The performance of ray tracing can be outstanding as long as detailed environment information is available. However, the computational complexity is regarded as a weakness of ray tracing even though multiple acceleration means have been proposed and experimented, such as bounding boxes to mitigate redundant imaging [44], preprocessing and discretization of the visibility tree [30, 74], leverage of multi-core processors [107] etc. Acceleration performances show a strong dependency on the objective scenarios.

By comparison, the ray launching is a transmitter-oriented brute force simulation. A large number of equally spaced rays are shot from the transmitter which represents the wavefront [66, 131]. A reception detection algorithm is designed to capture the rays being received [106]. Superposition of the arriving rays depicts the nature of constructive or destructive interference. In the process, reflection, diffraction and transmission behaviours of each ray is traced and recorded [64].

A ray is terminated if one of the fading-out criteria are satisfied. Ray launching usually does not consider scattering because specular reflections on a surface form another wavefront [10]. As long as the number of rays is adequate, different propagation behaviours of a wavefront upon a structure can be determined.

A significant advantage of ray launching lies in the propagation paths generated by an emitter being independent of the receiver. Therefore, an identical receiving mechanism can be assigned across a scenario to obtain channel parameters with respect to all receiver locations in one go. Furthermore, the intersection behaviours of rays during propagation are also independent which allows acceleration using parallel computing. In summary, although a brute force approach, better speed performance is anticipated using ray launching.

According to [87], convergence experiments show that with over seven orders of reflections the ray launching estimations broadly agree with measurements. Alwajeih *et al.* [9] found that ‘10R1D’, meaning 10 orders of reflections and one order of diffraction, were sufficient to deal with complex urban environments. As is known, the computational cost of ray tracing rises exponentially when the reflection order increases. Ray launching maintains a constant number of wavefronts thus suffering less from the increase of the reflection orders.

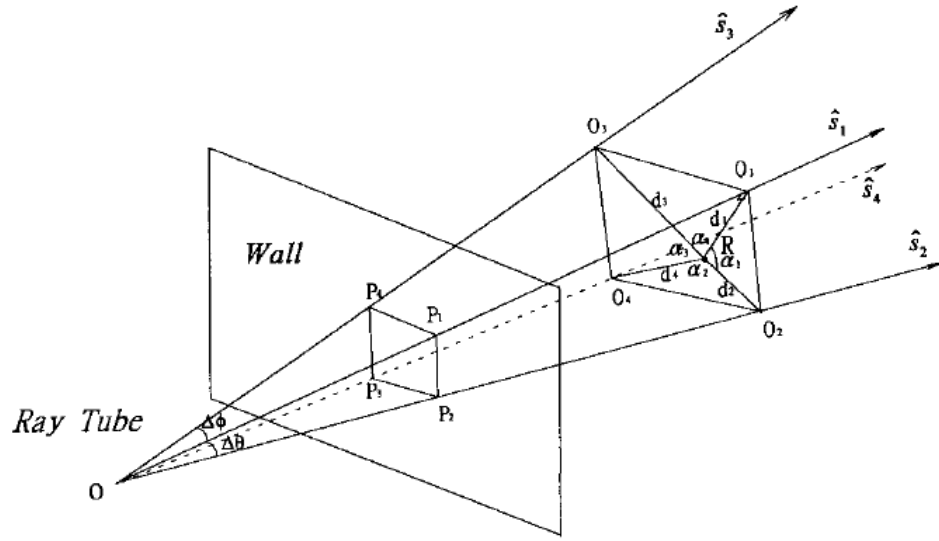


Figure 2-2: An illustration of ray tube intersection detection taken from [135].

The selection of model is indeed a trade off between expected accuracy and computational complexity. So far, deterministic propagation models are more reliable and accurate, and represent the future trend. Although a ray launching model is deemed superior in most aspects, it requires further simplification to suit large area urban environments. Researchers in the literature have attempted to improve ray launching accuracy and efficiency by applying tube-based ray shooting [100, 135]. This method regards a group of neighbouring rays as a ray tube which should see similar behaviour if intersecting with a common flat surface. Once a significant intersection geometrical difference is found within a tube, the boundary of a surface is detected. Therefore, the reflected wavefront can be defined by only considering the boundary tubes which reduces intermediate computations. However, the computation of intersection criteria is more complicated and even more dense rays are required. Consequently, the acceleration provided by ray tubes is questionable. An illustration of a ray tube is shown in Figure 2-2.

Azpilicueta *et al.* [12] proposed the use of an Artificial Neural Network to recognize the pattern of similar intersection behaviours, so as to speed up ray launching process in indoor applications. Agelet *et al.* [7] proposed to divide a scenario into bounding boxes to eliminate redundancy intersection searches as is illustrated in Figure 2-3. Although good performance is found using these methods, they greatly depend on the scenario being investigated.

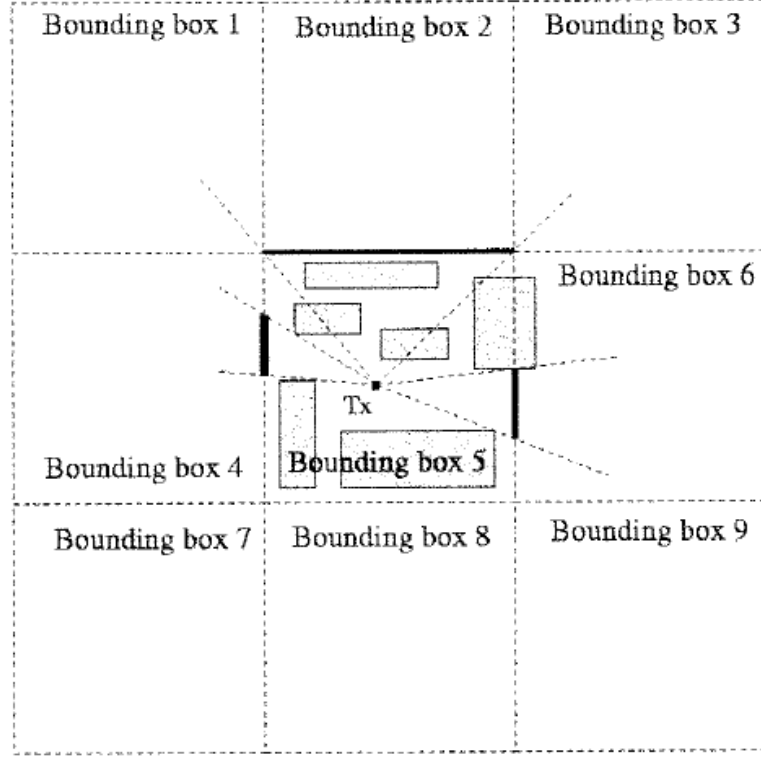


Figure 2-3: Dividing a scenario into bounding boxes to reduce computation complexity on intersection detections taken from [7].

Five propagation primitives are usually specified in a full-wave deterministic model, including free space propagation, reflection, diffraction, wall penetration and diffuse scattering [11]. Excluding diffuse scattering which is not necessary in ray launching, some propagation behaviour can be extremely expensive to simulate but making very limited contribution to the accuracy performance [68]. For example, modelling wall penetration not only demands precise information on thickness and material, but involves complicated polarization [86]. As a result, integrating semi-deterministic algorithms into the deterministic model provides a reasonable simplification [136].

The Friis transmission equation is regarded as the most popular formula to estimate the path loss as a function of propagation distance [105]. The received power,  $P_r$ , calculated using this formula can be written as;

$$P_r = \frac{G_t G_r \lambda^2}{(4\pi d)^2} \cdot P_t \quad (2.1)$$

where  $G_t$ ,  $G_r$  are transmitting and receiving antennas gains, respectively,  $\lambda$  denotes the wavelength,  $d$  is the cumulative travelled distance and  $P_t$  is the transmitted power. Al-



though phases of signals are omitted which results in a missing presentation of constructive or destructive interference, the Friis transmission equation is adequately accurate and extremely efficient in far field applications [102], for example urban propagation modelling where the scale is much larger than multiple wavelengths of opportunistic signals.

The reflection coefficient is defined to compute how much electromagnetic energy is dispersed in that ray with respect to incidence angle and surface material [76]. In terms of polarization, the vertically polarized component of the reflected ray is derived from the vertical form of reflection coefficient;

$$R_{vertical} = \frac{\varepsilon \cdot \sin \theta - \sqrt{\varepsilon - \cos^2 \theta}}{\varepsilon \cdot \sin \theta + \sqrt{\varepsilon - \cos^2 \theta}} \quad (2.2)$$

and the horizontally polarized component follows;

$$R_{parallel} = \frac{\sin \theta - \sqrt{\varepsilon - \cos^2 \theta}}{\sin \theta + \sqrt{\varepsilon - \cos^2 \theta}} \quad (2.3)$$

where  $\theta$  is the incidence angle,  $\varepsilon$  represents the electric permittivity of the surface material which can be obtained from;

$$\varepsilon = \varepsilon_r - j60\sigma\lambda \quad (2.4)$$

where  $\varepsilon_r$  is the relative electric permittivity,  $\sigma$  indicates the electrical conductivity,  $\lambda$  equals the wavelength.

As is widely acknowledged that UTD [33, 95], which stems from GTD [17, 54], provides a precise description of diffraction. Diffraction can be categorized into edge diffraction which generates a plane circle or cone-like diffracted scatterers, and pin point diffraction in which the pin point becomes a virtual source and diffuses the energy all over the space [60]. As a result, genuine ray launching modelling of diffraction means thousands of hybrid rays would have to be considered. Owing to a high diffraction loss, typically only one order of diffraction in urban environments is considered. Similarly, researchers in the literature also defined ‘diffractivity’ [11] to calculate edge-diffracted rays in a deterministic way.

In the case of long distance propagation, the most significant arrival is likely to come from multiple roof-top diffraction [137], however, the exact roof-top diffraction can be difficult to reconstruct. In order to ease the computation, a semi-deterministic roof-top diffraction model is adapted in Chapter 3 to simplify our 2.5D ray launching engine with little compromise in accuracy. These definitions of propagation mechanisms constitute the core of a ray launching engine.

Common 2D propagation models employ digital map databases containing geometrical data on objects, surface or edge coordinates and tree locations. Given the transmitter position and power, a cluster of rays are initiated with a specified increment angle. As each ray propagates in the environment its first incidence with surfaces, edges or the boundary of the defined area will be processed with the help of either Fresnel reflection equations [11], UTD or transmission models [69, 90]. Intersection detection which determines incidence points will not stop iterating until the ray strength fades below a threshold level, the intersection order reaches a specified limit, or shooting out of the zone of interest.

A ray-capturing mechanism is usually adopted to assist arriving detection, e.g., a reception sphere [106]. The electromagnetic field strength at the receiver is the sum of each ray that passes through the receiver position, i.e., superposition of all arriving rays.

3D wave propagation models could be extended from 2D models with selective considerations of interactions as well as amendment of coordinate system [62, 70, 71]. Firstly, ray vectors should be defined at the transmitting point in angular representation with designated rotation angle and elevation angle increment. Then data about object parameters should be stored, such as object positions, surfaces, edges or vertices functions, material conductivity, permeability, dielectric constants etc. Via ray vector intersecting with surfaces, edges or vertices functions, the travelled distance, incidence position and angle to the object as well as signal strength can be derived. The processor genre depends on which kind of incidence takes place, specular reflection, transmission, edge diffraction, or vertex diffraction [28, 74].

3D models are widely used in indoor environments at high frequency bands because more sophisticated multipath effect is anticipated in the channel. However, the computational cost is even higher as a price. An experimental 3D modelling attempt in Chapter 3 verifies this statement. 2.5D models, which only consider vertical domain diffractions and ground reflections on a basis of 2D models, are generally deemed as a practical solution for urban propagation modelling [65]. Literature reviews have demonstrated effectiveness of 2.5D models in different urban scenarios [39, 41].

Overall, wave propagation modelling is able to present dynamic channel characteristics as a function of spacial geometry. To be able to address the multipath phenomenon plays a growingly important role in modern wireless communication applications. The ever-increasing computation capability of processors and easy access to city layout databases are encouraging the widespread use of deterministic models.

## 2.3 Digital mapping

A successful application of propagation modelling in cities relies on an accurate environmental description. Nowadays digital map databases are prevailing for military, commercial, academic as well as public uses. 2D digital maps are easily accessible. Many government based and commercial institutions have developed comprehensive 3D building databases covering a vast majority of cities over the world. For instance, Google Earth stores terrain data from satellite imaging which is able to present coverage type, height above sea level as well as detailed shape of a building or structure. An aerial view of Hoxton district in London is captured from Google Earth shown in Figure 2-4.



Figure 2-4: An aerial view of Hoxton district in London captured from Google Earth.

Coast initiated a digital map project called OpenStreetMap in 2004 [2]. It constructs a free editable map of the world from volunteered geographic information. With over 2 million registered users, 2.5D data of most urban layouts, i.e., building latitude and longitudes as well as heights, are available and constantly updated. Quite a few researchers in the literature have adopted and validated the OpenStreetMap in the use of urban propagation modelling [88]. An advantage of using the OpenStreetMap for propagation modelling is that the database is constantly updating to approach a real-time reconstruction. Moreover, it is free.

A powerful user interface, named as Overpass turbo [3], is built upon the OpenStreetMap. It relies on a set of API queries to carry out data mining upon the interactive map. Depending on the user requirements, Overpass turbo returns specific

geographical information, such as public, commercial or residential buildings, vegetations, parks, rivers, motorways, railways etc., in digital formats. For example, a *kml* file can be generated and translated into structural arrays for Matlab. The object extraction process is detailed in Chapter 3.

Rich and detailed digital maps with high accuracy give a solid foundation for propagation modelling in urban environments. Compared to current indoor deterministic modelling where environmental information has to be specially entered and yet constantly changing, urban modelling remains effective for a longer period such that it possesses better temporal validity for practical use.

## 2.4 Fingerprinting

Laborious data collection and dynamics of the urban environments can be effectively addressed using deterministic propagation modelling. Given objects layout in the scenario and electromagnetic coefficients of materials, a combination of RSS, AOA and TDOA at known positions may provide evidence to predict locations in context of multipath [40]. In fact, RSS has been widely investigated and validated in indoor localization systems [18, 113, 132]. The major concern is that the number of opportunistic signals needs to be adequate to support a robust accuracy of analysis. Measurements of AOAs with coarse precision, e.g., using 'MUSIC' method [23, 116], are also capable of producing spatial information as long as multipath propagation is explicitly identified. Jin *et al.* [51, 110] proposed and experimented on using TOAs to discriminate emitter locations. Similarly, identification of the propagation path is the key to success.

A new concept called 'fingerprinting' in terms of channel characteristics has drawn increasing attention recently. Uniqueness is the most significant feature of a fingerprint. Also a fingerprint should possess a degree of tolerance such that slight distortions can be mitigated. This requires appropriate quantification of the fingerprints. Since spatial diversity has been broadly recognized in channel characteristics [34, 119], fingerprinting may be an approach to finding an access point or reference point locations. In the wider use of the term, a radio fingerprint can be defined as an eigenvector incorporating components such as RSS, AOA, TOA etc. Taking advantage of opportunistic signals, the eigenvector may be expanded to incorporate diverse channel characteristics. Given a reliable mapping function, such fingerprints might lead to correct geographical locations.

Fingerprint-based localization proposals are presented in [8, 22], where dominant AOAs, RSSs and TDOAs are registered as fingerprints. Very good accuracy was found through tests. Zou *et al.* [141] developed a novel approach which extracts location

characteristics by combining wavelet transformation and singular value decomposition. The eigenvectors derived are then used to build the fingerprint-location database. Simulation results exhibit very good localization accuracy.

Alternatively, unknown emitter locations may be revealed via fingerprinting because an emitter is generating different field observations at different locations. Coco *et al.* [20] proposed to find the most probable source location among potential grid points by solving a cost function established upon measured fingerprints. Ray tracing is the ideal tool for generating source fingerprints and good accuracy is seen from simulation. Phelan *et al.* [92] extracted characteristics, such as SNR (Signal to Noise Ratio), BER (Bit Error Rate), AOA and TOA, from channel to locate a radio source using a clustering algorithm. This method is validated through experiments. However, the user is required to obtain detailed information on what is being transmitted. Wadhwa *et al.* [126] designed and tested an interesting tracking algorithm utilizing an Unmanned Aerial Vehicle (UAV) to automatically approach the emitter in which RSS essentially makes the fingerprint. Although the algorithm is validated and optimized, it may take a long while for the UAV to arrive at the source.

## 2.5 Artificial neural network

A proper fingerprint definition is a *priori* to bond channel characteristics and locations, nevertheless, a robust matching algorithm referees the success of fingerprint-based localization systems. Linearity certainly does not apply to high-dimension fingerprint eigenvectors. Although non-linear regression methods are tested and proved working, very accurate measurements on TOA or AOA are critical to perform a valid location estimation [77, 103]. In this case, subtle uncertainties of temporal synchronization, variations in antenna radiation pattern, constructive or destructive interferences bring serious problems to the problem of localization accuracy.

In order to map high-dimensional discontinuous fingerprints with a mass of urban positions, machine learning, particularly Artificial Neural Network (ANN), accounts for a potential solution. Some researchers in the literature have succeeded in training passive radio measurements using ANN [127, 141]. A basic hypothesis for this approach is that the fingerprints extracted are unique and static to define locations.

ANNs are famous for handling discontinuous data. The more training resources there are the more trustworthy a mapping function would be. ANNs are comprised of neurons, connections and weights, propagation function and learning rule [1]. These parameters are all liable to user customization for an optimized performance.

An ANN can be generally regarded as a simple mathematical map relationship;

$$f : X \rightarrow Y$$

where  $X$  stands for a set of training inputs, and  $Y$  is usually called target. Once function  $f$  is agreed on convergence, it automatically generates recognitions on unknown input materials. Depending on learning paradigm used, ANN is divided into three types, including supervised learning, unsupervised learning and reinforced learning. Since the fingerprints are essentially labelled channel characteristics, it is supervised learning that best fits our application.

The core power of ANN comes from its ability to learn. The goal is basically minimizing a cost function which is chosen by the user. Back-propagation ANN is the most widely acknowledged learning pattern when used in conjunction with fingerprinting [124]. Back-propagation algorithm is a method to calculate the gradient of the loss function (produces the cost associated with a given state) with respect to the weights in an ANN. Sotiroidis [115] succeeded in mapping RSSs and TDOAs onto locations, using feed-forward back-propagation ANN which marked a milestone for fingerprint-based localization. It is also noted that the ANN produced smooth outputs of estimation given erroneous or out-of-bounds inputs which is ideal for localization purpose.

## 2.6 Pattern recognition

The reliability of ANN depends on a large pool of training resources. In terms of radio source fingerprinting, it takes a great deal of simulation effort for a ray launching model to obtain sufficient source fingerprints for training. In this case, applying pattern recognition techniques on easily accessible fingerprints could make a more practical solution.

Pattern recognition is a branch of machine learning that focuses on the recognition of patterns and regularities in data [73]. Labelled training data, or so called supervised learning, is usually the resource for pattern recognition. The processing on both labelled and unlabelled data is referred to as data mining, which is the transformation process of large data sets for better understanding involving machine learning, statistics, and database systems. Data mining, particularly pattern recognition, using fingerprints may be able to locate unknown RF sources. In [91], it can be seen that a diverse range of engineering problems are currently seeking for answers from pattern recognition.

The K-Nearest Neighbour (KNN) is one of the simplest but most popular data mining algorithms. Literally,  $k$  closest candidates are regarded as a cluster for classification and regression purpose. KNN is a type of instance-based learning where the function is

only approximated locally and all computations are deferred until classification [133]. When  $k$  equals 1, an intuitively closest match in the feature space is returned. This can be useful for source localization if the fingerprints are reasonably dispersed in the feature space.

KNN algorithm can be adapted to a weighted classifier which assigns a measure of distance to each of the  $k$  neighbours. Such classifier is also called K-Weighted-Nearest Neighbour (KWNN). KWNN allows simplistic study on distribution pattern of dense training samples. Huang *et al.* [45] used both KNN and KWNN as matching algorithms for RSS-based indoor localization and verified significant improvements using the latter over the former. Zou *et al.* [141] applied KWNN for recognition of extracted electromagnetic fingerprints where very good accuracy was found.

Dynamic Time Warping (DTW) is an alternative pattern recognition algorithm specially designed for time series to mitigate mismatching. Given two data series  $\vec{a} = (a_1, a_2, \dots, a_n)$  and  $\vec{b} = (b_1, b_2, \dots, b_m)$  of length  $n$  and  $m$ , respectively, an  $n$ -by- $m$  matrix can be constructed in which the  $(i, j)$  element contains the distance between two points  $a_i$  and  $b_j$ . Each matrix element corresponds to the alignment between points  $a_i$  and  $b_j$ . A warping path  $W$  is a contiguous set of matrix elements that define a mapping between  $\vec{a}$  and  $\vec{b}$ . The  $k$ -th element of  $W$  is defined as  $w_k = (i, j)_k$ . Hence, the DTW is the path that minimizes the warping cost given by

$$\text{DTW}(\vec{a}, \vec{b}) = \min \left\{ \sum_{k=1} w_k \right\}$$

If a complete spatial distribution of electromagnetic characteristics can be obtained, radio source localization becomes a deterministic and simple problem. Since LOS short-distance propagation always produces higher path losses than NLOS long-distance propagation, the peak of RSSs distribution, the valley of TOAs distribution, or the radiation center of the most significant AOA distribution on an aerial-view 2D heat map definitely represents the emitter location. As a result, it is straightforward to trace the radio source. However, precise and dynamic real-time channel monitoring of an urban scenario is not practical at all. Source localization essentially makes a compressed sensing problem.

Compressed sensing aims at an efficient reconstruction on undetermined linear systems. This is based on the theory that, via optimization, the sparsity of a signal can be used to recover it with far fewer samples than required by the Shannon-Nyquist sampling theorem [122]. It requires merely a fraction of outstanding samples to reveal the source location instead of the whole picture. This involves careful selection of the sampling strategy. Walter *et al.* [128] utilized compressed sensing to locate unknown

emitters with good accuracy found. Whereas, the power supply and maintenance bring new technical difficulties. Moreover, questions still remain where to put the sensors and how many of them are necessary. These also motivate us to use a UAV as remote sensor which is further explained in Chapter 5.

The various electromagnetic characteristics do not offer equally significant information in terms of localization capability. These labelled features can be transformed onto other labelled domains to reveal information from different perspectives. The Discrete Fourier Transform (DFT) is a lossless transformation from time domain to frequency domain. The  $k$ -th DFT coefficients of  $x$ ,  $X_k$ , can be expressed as

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-j2\pi kn/N} \quad (2.5)$$

where  $N$  is the length of samples and  $x_n$  represents the  $n$ -th sample value. Real and complex parts of  $X_k$  exhibit periodic features of  $x$ .

Labelled features can also be transformed onto unlabelled domains to present explicit mathematical significances, e.g., using Principal Component Analysis (PCA). PCA is a statistical procedure which uses an orthogonal transformation to convert a set of observations of potentially correlated variables into a set of values of linearly uncorrelated variables [117]. PCA is often used for the purpose of dimension reduction. Applying PCA on fingerprints may allow us to find the best coordinate system to differentiate them although physical meanings would be lost.

There are many other pattern recognition instances for solving localization problems. Jin *et al.* [51] successfully adopted extreme learning machine upon RSS variation gradient to predict radio source location. Coluccia *et al.* [21] integrated Maximum Likelihood into localization algorithms by minimizing a cost function which is transformed from RSS-location correlation matrix.

## 2.7 Parallel computing

Parallel computing is currently a hot topic of study which breaks a large serial task into sub-tasks and executes all concurrently. Parallelism can take different forms such as bit-level, instruction-level, data-level, and task-level parallelisms. Multi-core processors are the most commonly seen parallel computing application. For example, the dual-core and quad-core CPUs (Central Processing Unit), which are widely used in modern computers, allow much faster computations compared to a single core of identical specifications.

Recently the computation potential of general purpose Graphic Processing Units



(GPU) is being recognized. The GPU is now far more than an embedded device for display operations. Modern GPUs are equipped for double-precision mathematical operations in a parallel configuration which sees broader applications in smart computation. Open Computing Language (OpenCL) and Compute Unified Device Architecture (CUDA) are classic parallel programming platforms designed for GPUs. An NVIDIA GTX 280M GPU is shown in Figure 2-5. OpenCL is a high level programming language with comprehensive supporting libraries. However, the CUDA platform provides direct access to the GPU virtual instruction set and parallel computational elements, which is more convenient for senior users to customize kernel assignment [104].

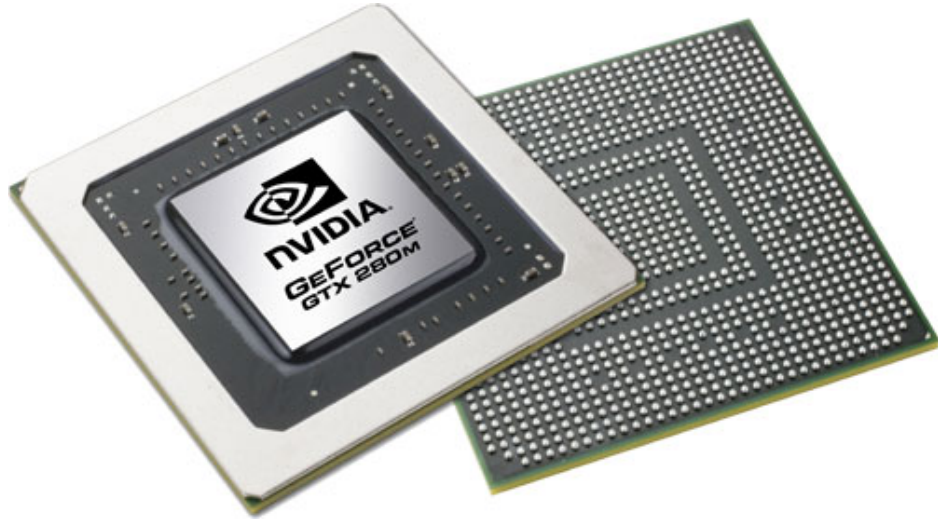


Figure 2-5: NVIDIA GTX 280M Graphic Processing Units taken from <http://www.nvidia.co.uk/page/home.html>

Although parallel computing is able to bring dramatic acceleration, it requires careful time and resource management on both host and device. OpenCL and CUDA programs are more difficult to write than sequential ones because concurrency introduces several new classes of potential software bugs [104], of which race conditions are the most common. Communication and synchronization between the different subtasks need significant work to get the best out of a parallel computing architecture.

As can be easily inferred, parallel computable subtasks should possess little dependency on each other during processing such that threads do not always have to hold for synchronization. There are actually multiple layers of independent computations in propagation modelling. For example, the channel characteristics between transmitter and receiver of different locations are derived individually, so are the propagation paths of the launched rays. The matching between the fingerprint candidates and each fingerprint in the database is also independent. Many researchers in the literature have

successfully applied GPU-based parallel computing to speed up wave propagation modelling [82, 107]. Acceleration of performance using CUDA with our model is further considered and demonstrated in Chapter 6.

## 2.8 Summary

This chapter provides a review of the existing propagation models and localization methods, as well as their limitations. In order to overcome these limitations, state-of-the-art propagation modelling techniques are then discussed on the possibility of aiding a fingerprint based localization solution. Data mining and machine learning can potentially be exploited to generate location decision algorithms. It is also noted that parallel computing architectures might be used to accelerate such a new localization scheme.

## Chapter 3

# Ray launching model

For the work in this thesis, a ray launching model has been developed and integrated from scratch. It is based on Matlab 2016b and consists of a few thousand lines of code. The model is expected to be able to simulate electromagnetic characteristics at different locations with respect to a radio source in a specified propagation environment. Although initially we considered a 3D ray launching model, our initial trial took a tremendously long time to process. Thus, the focus of this work has become a 2.5D model which proved to be robust in typical urban environments.

A traditional ray-type wave propagation model consists of three phases: initiation, ray shooting and field reconstruction; although different ray-type models may have more pre-processing steps [109]. The initiation phase aims to provide details on the electromagnetic properties of the environment and set up the propagation geometry of the transceivers. During the ray shooting phase, free space propagation applies to each ray unless intersection occurs. Depending on which kind of intersection is taking place, a reflection, diffraction or transmission processor is exploited to calculate the loss and redefine the ray that propagates forwards. Concurrently, a reception detection processor keeps a record of the rays that are arriving at the receiver. Eventually, the received signal properties are acquired from the superposition of all rays that fall into the reception sphere.

This 2.5D ray launching model has also been validated against COST 231 action measurements in the city center area of Munich from Mannesmann Mobilfunk GmbH [36]. The mean absolute error along any of the three measurement routes is less than 8 dB, which proves the effectiveness of this model.

### 3.1 3D modelling

Urban propagation environments are usually very complex. Various factors lead to a remarkable degree of uncertainty to accurately predicting the ray propagation behaviour, e.g., irregular geometric shapes of modern architectures, diverse surface materials, different kinds of vegetation, traffic and pedestrians [102]. A useful propagation modelling tool should be able to address the representations of these factors. Therefore, the first step is to decide the genre of modelling.

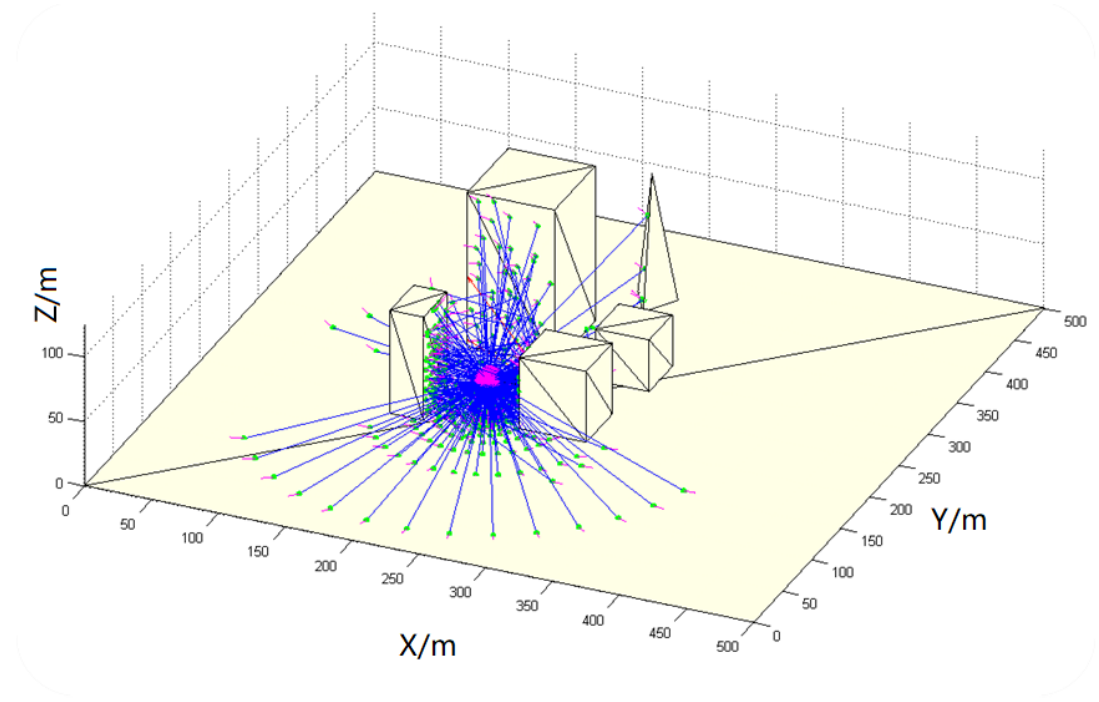


Figure 3-1: An attempt of 3D ray launching modelling in which buildings and ground are divided into triangles and coloured in yellow; blue lines represent LOS propagation paths or reflections; red lines represent diffracted rays; green dots are intersection points; magenta arrows represent rays terminated or without reception.

As has been discussed in Chapter 2, 3D ray launching models are among the most powerful deterministic models. We attempted to build a 3D ray launching model based on Matlab. A demo of the propagation paths in a virtual environment is shown in Figure 3-1. The ground is represented by a square plane of 500m side length. We defined four cubic buildings and a triangular pyramid with flat surfaces. A radio source is allocated at (200m, 200m, 20m) transmitting evenly distributed rays over a spheric surface. Note that all flat surfaces are split into triangles because they are easy for ray-surface intersection computations.

It can be seen that the ray coverage becomes sparse as they travel away from the source. This would likely result in missing intersections. The only solution is to increase the density of rays being shot which enormously increases the computational cost. Even though thousands of rays are launched from the source in this demo taking over 20 minutes to run, the 3D model detects few edge diffractions with close-by buildings. We suppose that hundreds of thousands of rays are necessary for the comprehensive modelling of urban scenarios. Such large computation time for only a modest increase in accuracy means that a 3D approach is not useful in this study. Hence, we determined to build a 2.5D model. The development process is detailed in the rest of this chapter.

## 3.2 Environmental entries

A detailed description of the environment and measurement devices is critical to the success of a wave propagation model. As has been stated previously, comprehensive digital maps of most cities are easily accessible. The urban propagation environment should be as fully defined as possible to ensure a close-to-reality propagation prediction. This section explains the preliminary works in the initiation stage.

### 3.2.1 Antennas and propagation media

Before initiating the ray launching model, the antennas and propagation media have to be specified. The transmitter and receiver locations also need to be known as *a priori* in order to perform reliable channel estimation. The antennas patterns can be generated and encapsulated as a function of departing or arriving angle. An example of a 2D radiation pattern is shown in Figure 3-2.

The media in urban scenarios can generally be regarded as air or free space. And the propagation in these environments usually satisfy far-field propagation condition [35]. This type of propagation within a homogeneous media always follows a straight line. Therefore, the Friis Transmission Equation can be applied to approximate the path loss due to free space propagation [105]. The derivation of received power,  $P_r$ , is given by Eqn. (3.1)

$$P_r = \frac{G_t G_r \lambda^2}{(4\pi d)^2} \cdot P_t \quad (3.1)$$

where  $G_t$ ,  $G_r$  are the transmitting and receiving antennas gains, respectively,  $\lambda$  is the wavelength,  $d$  is for the cumulative travelled distance and  $P_t$  is the transmitted power. Hence, the total travelled distance, departing angle from transmitting antennas and arriving angle at the receiving antennas should be recorded for each ray in order to

give a general path loss estimation. Since the wavefront is represented by a finite number of rays, the transmitted power of each ray is obtained from splitting the source power into the number of rays being shot.

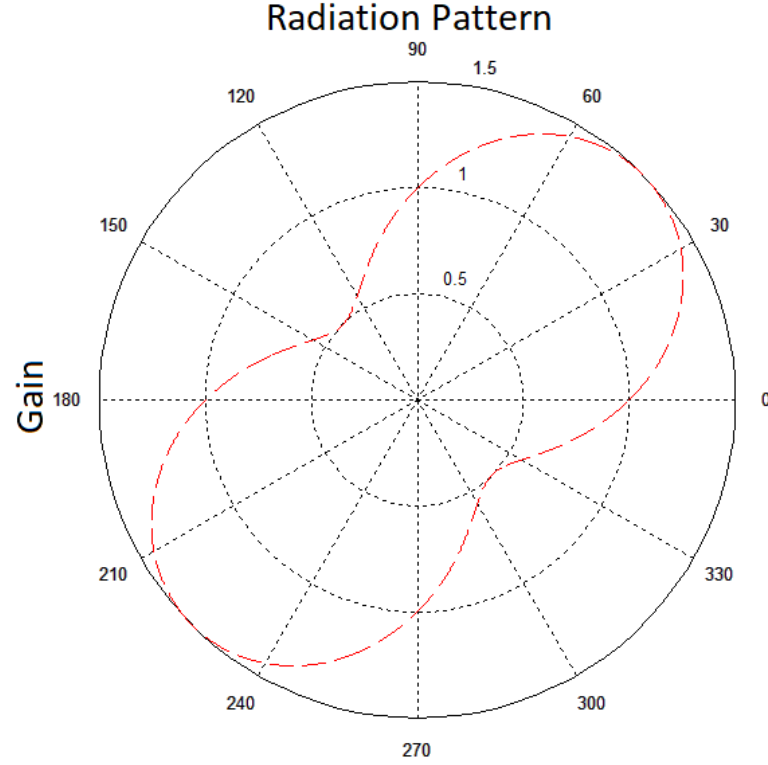


Figure 3-2: An example of antennas radiation pattern in horizontal domain, the dashed red envelope represents the gain as a function of angle in horizontal plane.

### 3.2.2 Urban layout entry

As has been introduced in section 2.3, the Overpass Turbo is able to provide an accurate, up-to-date and handy access to almost all major cities across the world.

We can locate a target area by running a query to search the name of that city or region, and zoom in or out to an appropriate size. By dragging a rectangular box to select the zone of interest, the buildings, structures and trees within the zone are highlighted. Notice that the absolute coordinates and heights of all objects in the zone are contained. We can then extract these data into a portable file, e.g., *kml* format, which incorporates global longitudes and latitudes of the structure vertices as  $\{lon, lat\}$  pairs.

The bottom left corner of the drag box should be specially marked as it will be used as the *Origin* in our ray launching simulator. If the global longitude and latitude of

this point are  $(O_x, O_y)$ , the Earth radius,  $r$ , at this latitude can be derived from Eqn. (3.2);

$$r = \sqrt{\frac{[r_{max}^2 \cos O_y]^2 + [r_{min}^2 \sin O_y]^2}{[r_{max} \cos O_y]^2 + [r_{min} \sin O_y]^2}} \quad (3.2)$$

given the earth maximum radius at the equator,  $r_{max} = 6378.137$  km, and minimum radius at either of the poles,  $r_{min} = 6356.752$  km. Subsequently, the relative coordinates of a point,  $P(p_x, p_y)$ , can be obtained via Eqn. (3.3);

$$(p_x, p_y) = [\sin(lon_i) - O_x, \sin(lat_i) - O_y] \cdot r \quad (3.3)$$

Following the above procedures, the structures on the map are transformed into polygons on a familiar Cartesian  $xy$ -coordinate system with the *Origin* located at the bottom left corner. The flat surfaces can then be extracted for ray intersection detection.

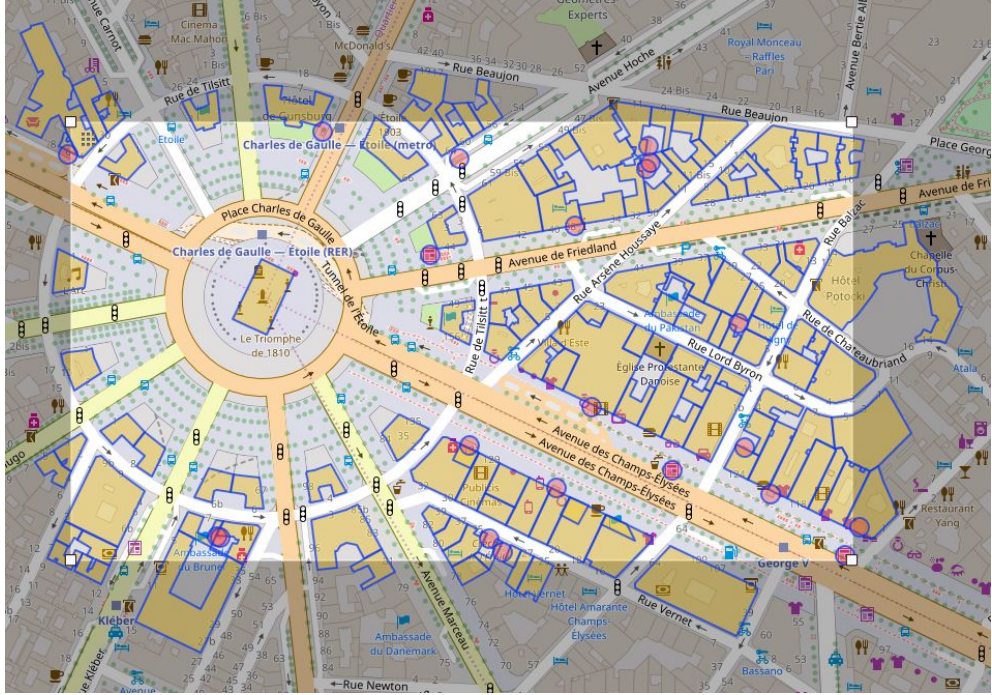


Figure 3-3: The Arc de Triomphe area, Paris, on the Overpass Turbo panel.

An example of the Arc de Triomphe area in Paris is shown in Figure 3-3. The highlighted buildings and structures are marked with a blue outline with bold yellow facets. Trees and smaller segments are covered by purple circles. The height of each structure is also stored in the file.



The extracted urban layout can be plotted in Matlab as displayed in Figure 3-4. A transmitter is added on the Champs Elysees Avenue marked as a red dot. A receiver is located on a nearby road. A series of rays are radiating uniformly from the transmitter marked as blue lines, which are subsequently reflected after intersection with flat surfaces. Obviously, the black polygons represent the buildings as viewed from above. The green circles are the trees. There are two red lines, one connecting a corner of the Arc de Triomphe and the transmitter, the other joining the same corner with the receiver. This is one of the diffraction paths being detected. A light blue line goes across the Arc de Triomphe to arrive at the receiver, which is a wall penetration instance. The diffraction and wall penetration are always tracing for each ray, but will only be displayed if the out-going ray reaches the receiver.

Note that only a limited number of rays are shot from the transmitter, which results in quite a few missing detections of apparent edge diffractions and transmissions. The coverage range of wave propagation is also restricted with areas of blind spot. This is primarily due to an insufficient number of rays being launched. In practical simulations, less than  $0.5^\circ$  of angular increment is usually allocated between neighbouring ray departures which generate over 720 rays. More densely shot rays guarantee more detailed intersection detections. However, as the number of rays increases the computation load also increases. Consequently, more comprehensive ray arrival information can be obtained at the receiver.

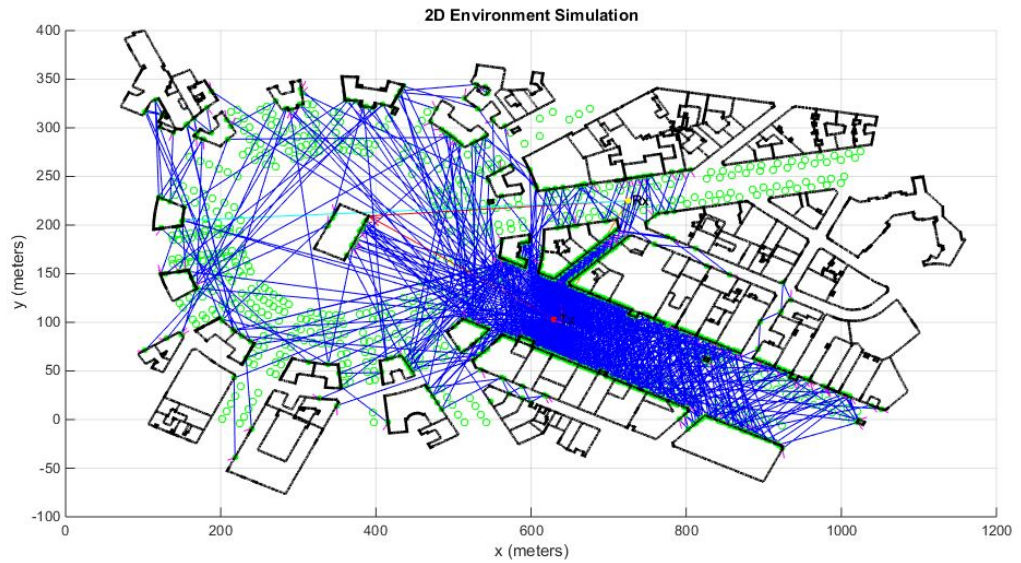


Figure 3-4: Propagation paths calculated using Matlab for the Arc de Triomphe area, Paris, given a virtual RF source at (630m, 100m). All reflected rays are presented in blue, whereas, only penetrated, diffracted and root-top diffracted rays that are received are shown in light blue, red and orange, respectively.



This simulation shown in Figure 3-4 at the Arc de Triomphe area takes over 1000 s, provided that only one transceiver channel is evaluated. Methods used to accelerate the simulator are detailed in section 6.

On the other hand, overly dense rays being shot means that double counting of rays is likely. When neighbouring rays going through an identical propagation path are received, all rays but one should be omitted since they represent a same wavefront. This selection function has been implemented by identifying rays that depart from a similar angle while going through the exact same sequence of surfaces and end up with similar travelled distances.

### 3.2.3 Trees

The modelling of wave propagation through vegetation can be very complicated. Firstly, in urban environments there are thousands of different kinds of trees with various shapes, heights and foliage densities. These properties usually change seasonally. The random layout of leaves and branches introduces complex diffractions, transmissions and scattering [38], which prevents a both accurate and efficient model.

Among a large number of studies on measurements and modelling of trees [96, 98, 114]. Benzair *et al.* provided a reliable empirical path loss model on wave propagation through trees [13]. This model considers the frequency (from 1-4 GHz), interception distance across the tree crown, as well as the season. The path loss formula is provided in Eqn. (3.4);

$$L_{tree} = a \times f^b \times d_f \quad (3.4)$$

in which  $f$  is the frequency in GHz,  $d_f$  is the tree depth traversed by the ray,  $a$  and  $b$  are constants from the lookup Table 3.1, which vary depending on seasonal foliage density as well as humidity.

Table 3.1: Table on parameters  $a$  and  $b$  of the tree loss model

Season	Summer			Winter		
Humidity	Median	50%	90%	Median	50%	90%
$a$	0.57	0.71	0.78	0.36	0.52	0.59
$b$	0.60	0.47	0.42	0.43	0.29	0.25

We use this model to predict the path loss through trees. The scattered rays by the tree are neglected since the power of these rays is so small that tracing them adds much computational burden but adds little to the accuracy [98]. In our 2.5D ray launching

model, these trees of which the precise locations can be accessed from the OSM are always treated as a cylindrical attenuating media with 3m radius. Hence, rays going through trees will be subject to power loss but do not change direction. An example of the transmission through a tree is shown in Figure 3-5.

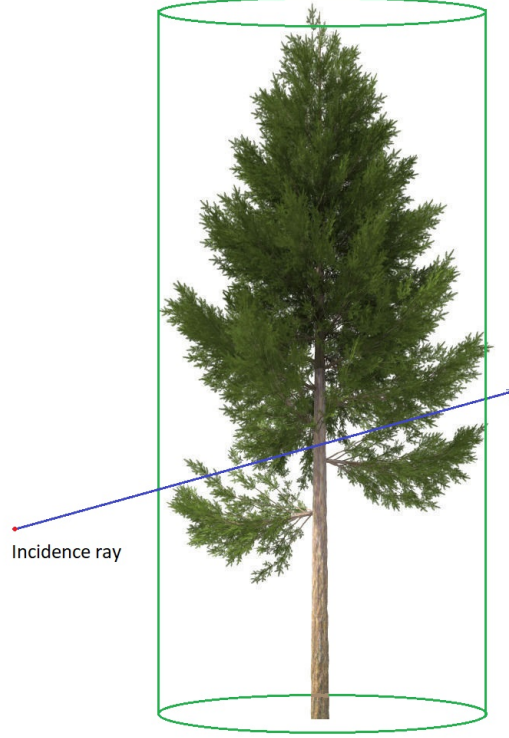


Figure 3-5: Transmission of a ray through a tree in which the crown is treated as a cylindrical attenuating medium.

### 3.3 Propagation primitives

Four propagation primitives are considered in the ray launching model including reflection, edge diffraction, wall penetration and roof-top diffraction. It is believed these propagation phenomena make the most significant difference to channel estimation [11]. These models are integrated in a variety of highly recognized models and measurements found in the literature. The mechanisms and empirical models in these primitives are explained in detail in this section.

### 3.3.1 Reflection

In urban environments, reflection plays one of the most significant roles which gives rise to multipath propagation. If the LOS path is blocked, the physical channel between a transmitter and a receiver is usually made of reflections from flat building surfaces. Studies of corridor models show typical propagation environments in the street canyons when both the transmitter and receiver are below the roof-top height [52]. Bertoni and Walfisch-Ikegami models [105], in which the signals are diffracted at the roof edges, also investigate the reflection phenomenon of rays in the vertical plane. In these cases, it is critical to determine the reflection mechanism of microwaves for building surfaces.

The frequency range of modern communication systems, e.g., radio broadcasts, DAB radio, GSM and Wi-Fi, is usually between 1MHz and 10GHz [94, 111]. The wavelengths of such signals are greater than 3cm. The modern building surfaces are mostly made of concrete, glasses, aluminium alloys or plastics, of which the surface roughness can be neglected compared to the wavelengths. Therefore, rough surface scattering is not taken into account. Hence, specular reflection generally applies to microwave intersections with building surfaces [67]. However, the polarization of the microwave after reflection varies depending on the surface material as well as the incidence angle.

According to the UTD, a vertically polarized wave generates both vertically polarized and horizontally polarized waves after reflection as long as the incidence angle is within  $0^\circ$  to  $90^\circ$  [95]. Note that the incidence angle here indicates the angle between the projection ray and the flat surface. However, the vertically polarized wave comprises most of the reflected energy as stated in [75, 86]. In the 2.5D ray launching model, we assume that rays launched from the transmitter are vertically polarized, and horizontally polarized waves after reflection can be neglected.

Specular reflection dictates that the reflection angle is always equal to the incidence angle within the intersection plane so that the direction of reflected ray can be determined. A reflection situation is demonstrated in Figure 3-6. A scalar, called the reflection coefficient, is used to describe how much electromagnetic energy is dispersed in that ray with respect to incidence angle and surface material [76]. According to the assumptions of our model, the reflected ray of a vertically polarized incident wave is also vertically polarized, and the reflection coefficient as follows Eqn. (3.5);

$$R_{vertical} = \frac{\varepsilon \cdot \sin \theta - \sqrt{\varepsilon - \cos^2 \theta}}{\varepsilon \cdot \sin \theta + \sqrt{\varepsilon - \cos^2 \theta}} \quad (3.5)$$

where  $\theta$  is the incidence angle,  $\varepsilon$  represents the electric permittivity of the surface material which can be obtained from Eqn. (3.6);

$$\varepsilon = \varepsilon_r - j60\sigma\lambda \quad (3.6)$$

where  $\varepsilon_r$  is the relative electric permittivity,  $\sigma$  indicates the electrical conductivity,  $\lambda$  equals the wavelength.

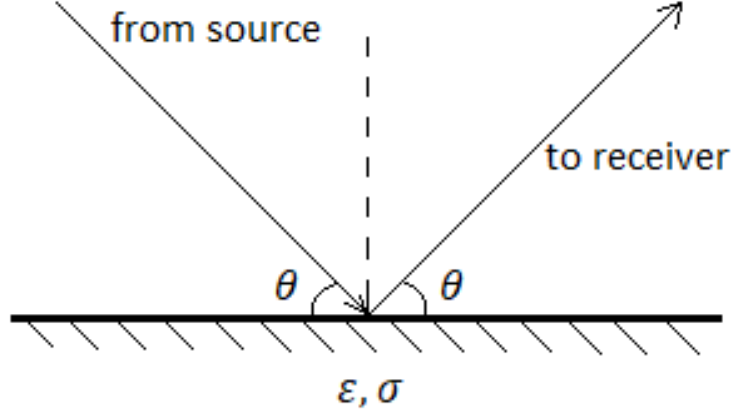


Figure 3-6: Reflection geometry on flat surface.

It is apparant that the electrical properties of the surface materials should be known as *a priori*. Provided the locations of buildings from digital map databases, we can assign surface material properties to each structure depending on the data available or simply assuming a common material. For instance, the relative dielectric constant and conductivity of concrete are 4.0 and  $10^{-4}$  S/m [93], respectively. They can be assigned to an unknown surface if corresponding information is not available since concrete is typically the most common surface material in urban environments.

Once a propagation ray intersects the surface, the reflected direction and signal strength of the reflected ray can be determined. The reflected ray makes a new virtual source and can continue to be traced onwards until it satisfies one of the fading-out criteria, including shooting out of the zone of interest, getting reflected for more than seven times, or attenuating below -140 dBm. Notice that convergence tests suggest that rays reflected more than seven times make little difference to the channel estimation [12]. If a ray is reflected over seven times it is regarded as terminated. Furthermore, it is difficult for mobile measurement devices to detect signals below -140 dBm.

### 3.3.2 Diffraction

Diffraction occurs when a ray hits the edge or cusp of a structure. In densely built-up environments, the vertical edges of buildings act as important media for RF signals to propagate along street canyons. In the Bertoni and Walfisch-Ikegami models [105], the horizontal roof edges illuminate the NLOS areas between rows of buildings. The tip of an object may also give rise to a spherical diffraction.

The GTD is regarded as a benchmark interpretation of the diffraction mechanism [54]. Depending on different incidence angles of a ray, it could generate a cone-like wavefront or a 2D circular wavefront normal to the edge [60]. A cusp of an object basically disperses the ray in all directions. The power of the diffracted ray is a function of several factors including arrival angle, departure angle, the distance from the source to the edge, the distance from edge to the receiver and reflection coefficients of the two surfaces that form the edge.

According to the literature on the validation of diffraction [9, 85], one order of diffraction is sufficient in urban environments due to the huge computational effort using higher order diffractions for only a limited increase in accuracy. For our 2.5D ray launching model, we assume the vertical edge and cusp of structures become new virtual sources and disperse the ray in all directions. Similar to reflection, we define a scalar, called *diffractivity*, to measure the power ratio of the out-going ray compared to the incident ray.

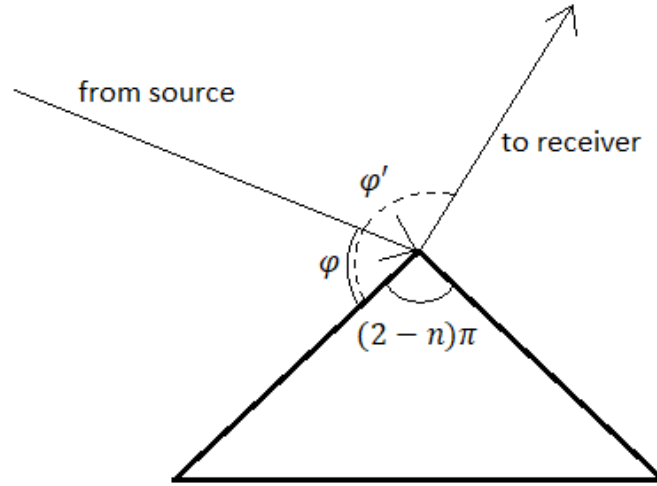


Figure 3-7: Diffraction geometry on the edge of a wedge.

Figure 3-7 shows a ray diffracted at the edge of a wedge in which  $n$  is the field angle. Thus, the interior angle can be expressed as  $(2 - n)\pi$ . Term  $\varphi$  is the incidence angle with respect to the incidence wedge facet, and  $\varphi'$  is the departing angle with respect to the same facet. According to [11], diffractivity can be obtained from Eqn. (3.7);

$$\begin{aligned}
D = \frac{-e^{-j(\pi/4)}}{2n\sqrt{2\pi}\beta} \cdot \{ & \cot\left(\frac{\pi + (\varphi - \varphi')}{2n}\right) \cdot F[\beta La^+(\varphi - \varphi')] \\
& + \cot\left(\frac{\pi - (\varphi - \varphi')}{2n}\right) \cdot F[\beta La^-(\varphi - \varphi')] \\
& + R_0 \cdot \cot\left(\frac{\pi - (\varphi + \varphi')}{2n}\right) \cdot F[\beta La^-(\varphi + \varphi')] \\
& + R_n \cdot \cot\left(\frac{\pi + (\varphi + \varphi')}{2n}\right) \cdot F[\beta La^+(\varphi + \varphi')] \}
\end{aligned} \tag{3.7}$$

where term  $\beta$  represents the wave number,  $R_0$ ,  $R_n$  refer to the reflection coefficients of the incidence wedge facet and opposite wedge facet. The reflection coefficients can be calculated using Eqn. (3.5) with respect to the incident wedge facet and opposite wedge facet, respectively. The Fresnel integral  $F(x)$  is given by;

$$F(x) = 2j\sqrt{x}e^{jx} \int_{\sqrt{x}}^{\infty} e^{-j\gamma^2} d\gamma \tag{3.8}$$

and  $L$  is a distance term in 2D scenarios defined as;

$$L = \frac{s \cdot s'}{s + s'} \tag{3.9}$$

where term  $s'$  denotes the distance from source to the diffraction edge, and  $s$  is the distance from the edge to the receiver. The angle terms  $a^\pm$  are given by;

$$a^\pm(\varphi \pm \varphi') = 2 \cos \left[ \frac{2\pi n N^\pm - (\varphi \pm \varphi')}{2} \right]^2 \tag{3.10}$$

in which the integers  $N^\pm$  are those which most closely satisfy Eqn. (3.11) and Eqn. (3.12);

$$2\pi n N^+ - (\varphi \pm \varphi') = \pi \tag{3.11}$$

$$2\pi n N^- - (\varphi \pm \varphi') = -\pi \tag{3.12}$$

### 3.3.3 Wall penetration

The concrete walls and glasses may also allow penetration of microwaves. Many measurement and modelling attempts have been made to describe RF transmission mechanisms through walls. Ding *et al.* [69] carried out wall-penetration measurements at 1.93 GHz. Okvist *et al.* [89] carried out measurements at 15 GHz for 5G application. Pena *et al.* [90] worked on 900 MHz wall-penetration tests and developed a statistical model. Jiang *et al.* [50] put forward a specific air-to-concrete transmission model. Deterministic models usually take into account the thickness of a wall and determine a refraction path within the wall. Multiple reflections and transmissions within the wall layer can also be elaborated as shown in Figure 3-8.

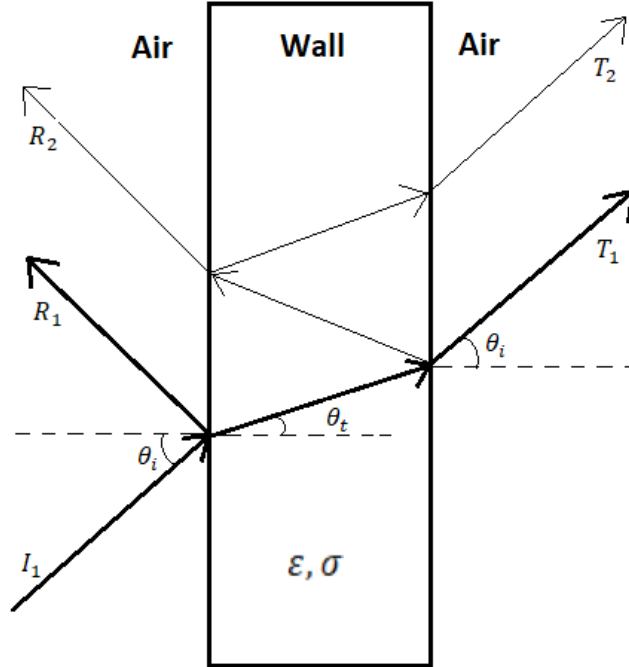


Figure 3-8: Comprehensive transmission geometry for modelling wall penetration.

However, the computational effort of many of these models is massive, and the directional shift due to refraction is often very small in urban scenarios. Moreover, the information on material and thickness which can be difficult to obtain and are essential to build such models.

The researchers in [111] have put forward a semi-deterministic model which approximates losses through one layer of wall regardless of the thickness. This model considers the frequency of the signal and its incidence angle which prove to be the most significant variables regarding the penetration loss.

The loss with respect to the frequency of the incident signal is provided in Eqn. (3.13);

$$L_{concrete,dB} = 4f_{GHz} + 5 \quad (3.13)$$

where  $f_{GHz}$  is the frequency in GHz (in the range of 0.1-60 GHz). The loss as a function of incidence angle is derived from Eqn. (3.14);

$$L_{angle,dB} = 20 \times (1 - \cos \theta)^2 \quad (3.14)$$

where the incidence angle is denoted as  $\theta$ .

The loss is presented in dB units. Hence, the signal strength drops by  $L_{concrete,dB} + L_{angle,dB}$  after each penetration without changing direction. In order to integrate this model into our ray launching engine, the number of transmission is recorded for each ray along its propagation path, and the losses will be deducted from the RSS during field reconstruction phase.

Up to three layers of wall penetration are allowed as vast attenuation is estimated for more than that. Note that reflection on the indoor surfaces for the penetrated ray is also a concern because in many cases the inner surface makes a multipath component before the rays arrive at the receiver. Since the indoor configuration can be too complex to model and the rays may find unexpected scatterings, only one order of inner surface reflection is taken into account where the order of penetration before or after indoor reflections is also restricted to one.

### 3.3.4 Roof-top diffraction

As the scale of the propagation domain increases, reflections, vertical edge diffraction and wall penetrations below the roof level become less detectable [59]. Besides the free space propagation loss, higher orders of these propagation behaviours result in the power degrading exponentially. The roof-top diffraction becomes dominant in such long distance wave propagation paths.

The propagation over roof tops is indeed very common in urban environments. For example, the LOS path from a base station to our mobile communication devices is always blocked in high-rise areas. A street view of New York city is shown in Figure 3-9. Thanks to the horizontal edges of buildings, the signal is able to perform multiple



forward diffractions past rows of buildings and go down from the nearest roof edge to the street [80]. This process can be divided into three stages, from the emitter to the first diffraction edge, multiple diffractions past building tops and from the last roof edge to the mobile.

Many researchers in the literature have shown measurement results at different frequencies in different cities, and proposed some empirical and semi-deterministic models to estimate the path loss [65, 81, 120]. Xia *et al.* presented an analytical model [134] which categorized such propagation behaviours into three circumstances, when the emitter is: above the roof-top height in Eqn. (3.24); about the roof-top height in Eqn. (3.25); below the roof-top height in Eqn. (3.26). It considers the rows of building as multiple knife-edge screens [27, 47, 121]. A diagram illustrating a below roof-top case can be seen in Figure 3-10.



Figure 3-9: New York street canyon view taken from: [www.ecology.com/2013/01/03/green-walls-cut-street-canyon](http://www.ecology.com/2013/01/03/green-walls-cut-street-canyon)

The model in [134] considers a variety of factors such as heights of the transmitter and receiver, distance from transmitter to the first diffraction edge, distance from the nearest roof edge to the receiver, travelled length horizontally, average separation distance between rows of buildings etc. This model is relatively accurate, and at the

same time, computationally efficient. The roof-top diffraction is mainly based upon this model.

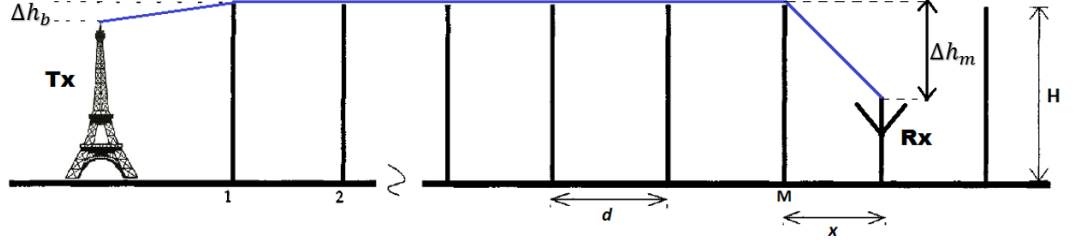


Figure 3-10: The geometry of roof top diffraction when the transmitter height is below average building height.

The propagation loss in the last stage, where rays drop from roof edge to the street, is given by Eqn. (3.15);

$$L_{rtd} = -10 \log \left[ \frac{\lambda}{2\pi^2 r} \left( \frac{1}{\theta} - \frac{1}{2\pi + \theta} \right)^2 \right] \quad (3.15)$$

where  $\lambda$  is the wavelength. The receiving angle in the vertical plane,  $\theta$ , and the distance from the edge to the receiver,  $r$ , can be expressed as follows;

$$\theta = \tan^{-1} \left( \frac{\Delta h_m}{x} \right) \quad (3.16)$$

$$r = \sqrt{(\Delta h_m)^2 + x^2} \quad (3.17)$$

where  $\Delta h_m$  is the height difference between the nearest roof edge and the receiver, and  $x$  is their horizontal distance.

The multiple screen diffraction loss can be expressed as;

$$L_{md} = -10 \log Q_M^2 \quad (3.18)$$

where variable  $Q_M$  is given by;

$$Q_M = \sqrt{M} \sum_{q=0}^{\infty} \frac{1}{q!} (2g\sqrt{j\pi})^q I_{M-1,q} \quad (3.19)$$

where  $M$  is the number of diffracting screens. Given in Eqn. (3.20), term  $g$  is defined as;

$$g = \Delta h_b \frac{1}{\sqrt{\lambda d}} \quad (3.20)$$

where  $\Delta h_b$  denotes the height difference between the transmitter and the first edge, and  $d$  indicates the average separation distance between rows of buildings.

In Eqn. (3.19)  $I_{M-1,q}$  satisfies a recursion relation as follows;

$$I_{M-1,q} = \frac{(M-1)(q-1)}{2M} I_{M-1,q} + \frac{1}{2\sqrt{\pi}M} \sum_{n=1}^{M-2} \frac{I_{n,q-1}}{\sqrt{M-1-n}} \quad (3.21)$$

with initial terms equal to;

$$I_{M-1,0} = \frac{1}{M^{2/3}} \quad (3.22)$$

$$I_{M-1,1} = \frac{1}{4\sqrt{\pi}} \sum_{n=0}^{M-1} \frac{1}{n^{2/3}(M-n)^{2/3}} \quad (3.23)$$

The path loss derivation can be simplified using a recursive approach considering building heights are uniform [134]. Given the total travelled distance from transmitter to receiver,  $R$ , the formulas used to determine propagation loss according to transmitter height at roof-top level, above roof-top level and below roof-top level are listed in Eqn. (3.24), Eqn. (3.25) and Eqn. (3.26), respectively;

$$L_{at} = -10 \log \left\{ \left( \frac{\lambda}{2\sqrt{2\pi}R} \right)^2 \cdot \left[ \frac{\lambda}{2\pi^2 r} \left( \frac{1}{\theta} - \frac{1}{2\pi + \theta} \right)^2 \right] \cdot \left( \frac{d}{R} \right)^2 \right\} \quad (3.24)$$

$$L_{abv} = -10 \log \left\{ \left( \frac{\lambda}{4\pi R} \right)^2 \cdot \left[ \frac{\lambda}{2\pi^2 r} \left( \frac{1}{\theta} - \frac{1}{2\pi + \theta} \right)^2 \right] \cdot (2.35)^2 \cdot \left( \frac{\Delta h_b}{R} \sqrt{\frac{d}{\lambda}} \right)^{1.8} \right\} \quad (3.25)$$

$$L_{bl} = -10 \log \left\{ \left( \frac{\lambda}{2\sqrt{2\pi}R} \right)^2 \cdot \left[ \frac{\lambda}{2\pi^2 r} \left( \frac{1}{\theta} - \frac{1}{2\pi + \theta} \right)^2 \right] \cdot \left[ \frac{d}{2\pi(R-d)} \right]^2 \cdot \frac{\lambda}{\sqrt{(\Delta h_b)^2 + d^2}} \cdot \left( \frac{1}{\phi} - \frac{1}{2\pi + \phi} \right)^2 \right\} \quad (3.26)$$

We integrate these models in our ray launching engine. In a 2D horizontal domain, a straight line connecting directly from transmitter to receiver is regarded as the propagation path. If the number of intersections with surfaces along this path is  $K$ , the

integer of knife edges approximates  $(K - 1)/2$ , which will be used as  $M$  with regard to the model (Eqn. (3.19)). In vertical domain, the total travelled distance is comprised of the distance from transmitter to the first edge, and then to the last edge, then to the receiver. The average separation distance between buildings can be easily obtained by dividing the horizontal distance by  $M$ . Depending on the height of transmitter compared to average roof-top height in the area, one of the formulas will be adopted to calculate the path loss.

### 3.4 Field reconstruction

The ray launching models rely on a geometrical capturing mechanism to decide whether a ray is truly arriving at the receiver or not. Using a reception sphere is a commonly acknowledged method in the literature [66, 131].

The propagation paths in a specific environment are solely determined by the transmitter location and the object layout. In other words, we assume that the receiving antennas, including users, are making little difference to electromagnetic fields.

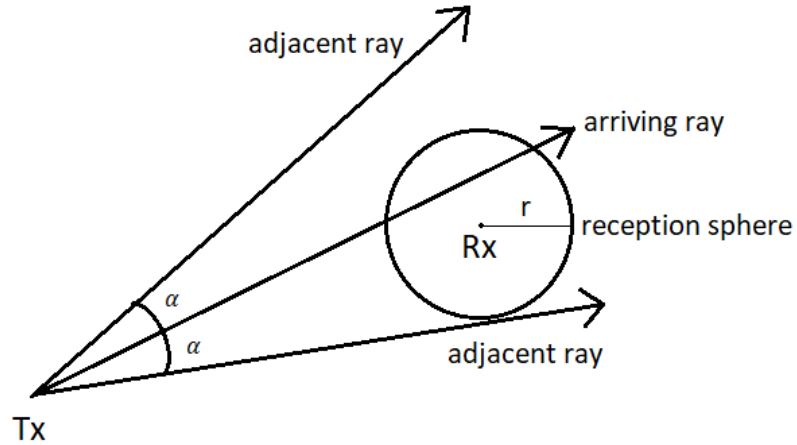


Figure 3-11: An illustration of the reception detection mechanism in which a virtual reception sphere is built at the receiver.

A reception sphere is a virtual circle centered at the receiver. The radius of the sphere is specified depending on the density of rays in the scenario. A search of the literature suggests that the scale of a reception sphere should be appropriate to only accommodate one ray at a time [100] as shown in Figure 3-11. This means that the radius of a sphere varies from location to location as neighbouring rays diverge when

they depart farther from the transmitter. In our model, a uniform radius is set in which double counted neighbouring rays are discriminated and treated as one capture.

Multipath propagation is very common in urban scenarios, which can be realized and analysed by wave propagation models. A reception sphere regards any propagating ray that falls within the circle as being received. The channel estimation is implemented by superposition of all rays that are deemed to arrive.

A single arrival may come from a combination of reflections, vertical edge diffractions, transmissions or directly from the nearest roof top. Three characteristics that are of our main interest are the arriving angle, delay and power. The arrival angle can be easily readable from the capturing geometry at the reception sphere.

The time delay, TOA, is approximately the total travelled distance,  $S$ , which should be carefully recorded, divided by the speed of light,  $c$ . The derivation of time delay can be written as;

$$\text{TOA} = \frac{S}{c}$$

With respect to [6, 11], the expression for computing the received power can be written as Eqn. (3.27);

$$P_r = P_t \cdot \frac{G_t G_r \lambda^2}{(4\pi d)^2} \left[ \prod_j R_j \right]^2 \left[ \prod_j D_k \right]^2 - 10^{-L_T/10} - 10^{-L_{rtd}/10} \quad (3.27)$$

where  $G_t$ ,  $G_r$  are the transmitting and receiving antennas gains, respectively,  $R_j$  represents the reflection coefficient of the  $j$ th intersecting surface, and  $D_k$  refers to the diffractivity of the  $k$ th diffracting edge. Terms  $L_T$  and  $L_{rtd}$  are transmission loss and roof-top diffraction loss, respectively.

The received characteristics allow us to determine the Power Delay Profile [57, 138]. A PDP is able to exhibit the received power of each ray versus the arriving time individually. It collectively depicts the channel impulse response.

The ray launching model is also capable of identifying directions of arrival with respect to the power. A polar coordinate plot, also called Angular Spread, at the receiver is used to specify the AOAs [23, 61, 116]. These arrivals can also be traced back to the source where rays departing from the very first wavefront are recognized. A polar coordinate plot at the transmitter shows in which directions of departures arriving detections are made at the receiver. Lastly, the radiation patterns of the antennas should be taken into account (Figure 3-2).

A virtual environment consisting of an enclosed box is designed to demonstrate how field reconstruction works (Figure 3-12). Seven objects of arbitrary shape are put in

an enclosed box which is 120m×120m square. Given the transmitter marked as a red dot, the receiver is a yellow dot positioned in the LOS area. The blue segments are either rays being shot or their reflections within seven orders of reflection. The magenta arrows refer to those rays fading out after seven reflections. Several red lines mark the diffraction paths. Wall penetration cases that are being received are indicated with light blue lines. Roof-top diffraction does not apply to this case.

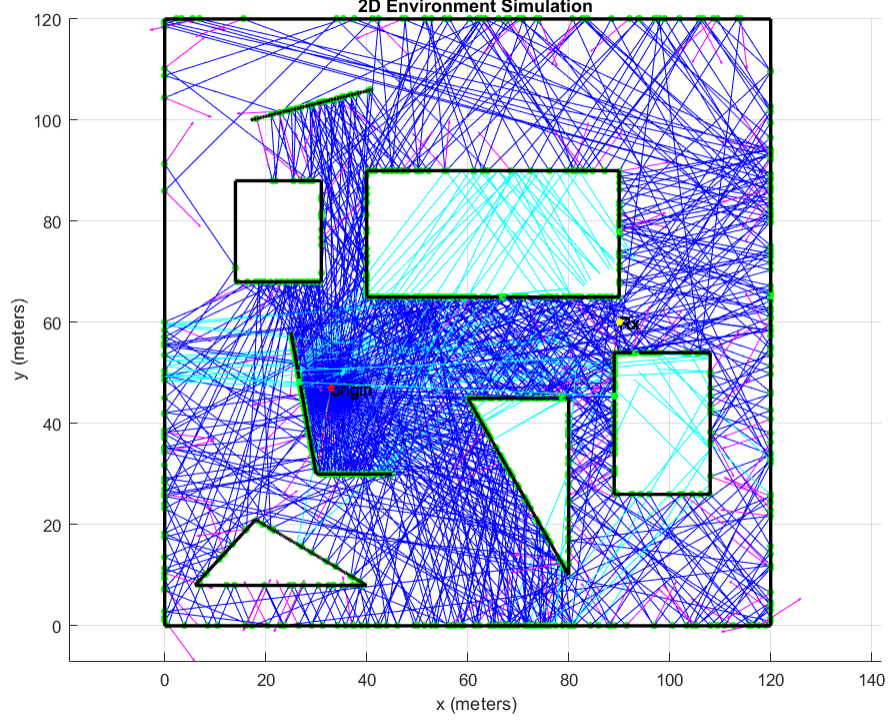


Figure 3-12: Propagation paths of an imaginary emitter at (33m, 47m), based on a virtual enclosed box scenario with seven arbitrary objects inside. All reflected rays are presented in blue, whereas, only penetrated and diffracted rays that are received are shown in light blue and red, respectively.

With the help of the reception detection function, receiving information can be gathered. Correspondingly, a PDP and two Angular Spreads are displayed in Figure 3-13 and Figure 3-14. Notice that the impulses in PDP are coloured with respect to their types of propagation. The earliest arrival is obviously the LOS ray with the largest signal strength, at approximately -75dBm. The very last arrival takes almost 0.85ms, which suggests a total travelled distance around 255m within the enclosed box.

The spikes in the Angular Spreads are labelled in dB relative to the minimum signal strength detected. It can be easily seen that most arrivals are clustered within a 180° sector of the receiver, which is intuitively recognizable given the geometry. On the

other hand, those arrivals may come from any direction from the original wavefront at the source.

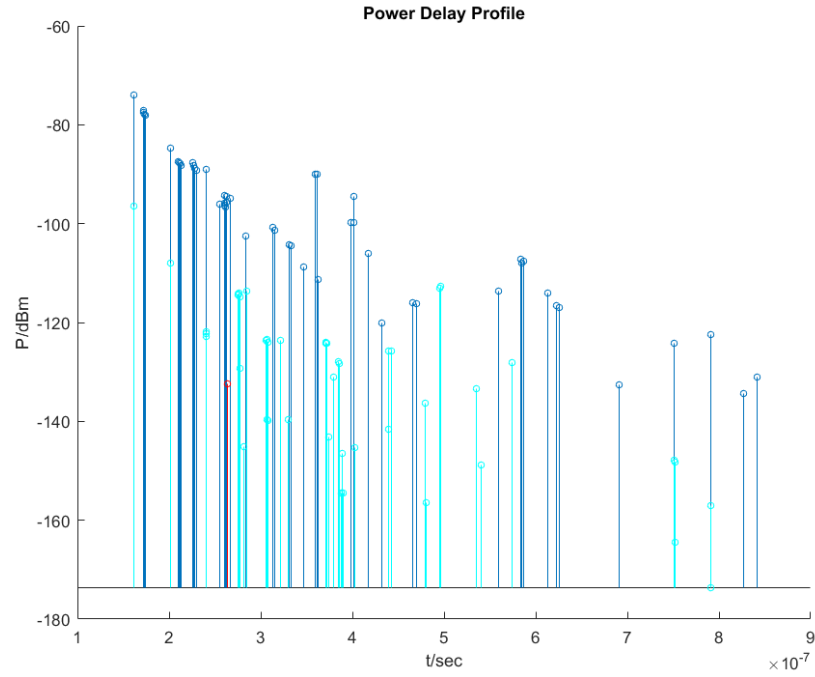


Figure 3-13: Power Delay Profile of the receiver at (90 m, 60 m) in the seven-object scenario.

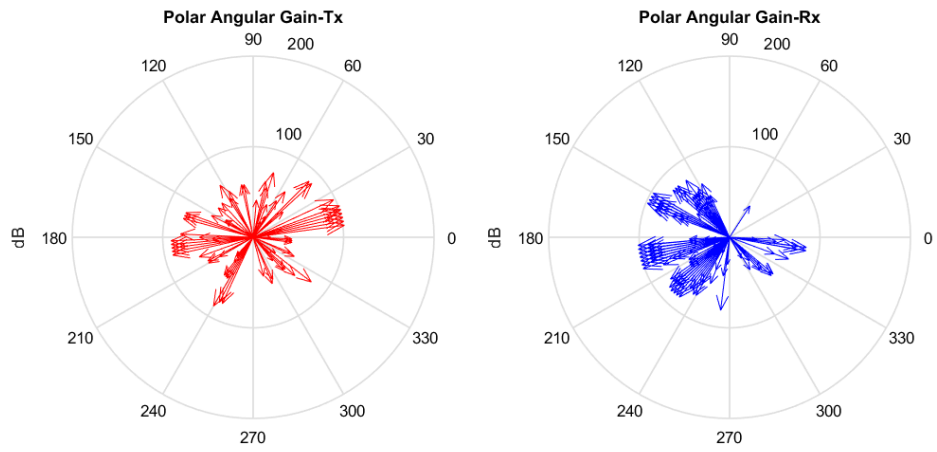


Figure 3-14: Angle of Departure and Arrival from the perspectives of transmitter and receiver, respectively, in the seven-object scenario. The receiver is located at (90 m, 60 m).



Generally, the PDP and the Angular Spreads present a realistic estimation of the received power, time delay of arrivals and angle of arrivals. Given the environment settings, the ray launching model successfully estimates electromagnetic features with respect to the transmitter and receiver locations. In this work, we consider how those features can be exploited to aid navigation and source localization.

### 3.5 2.5D ray launching model validation

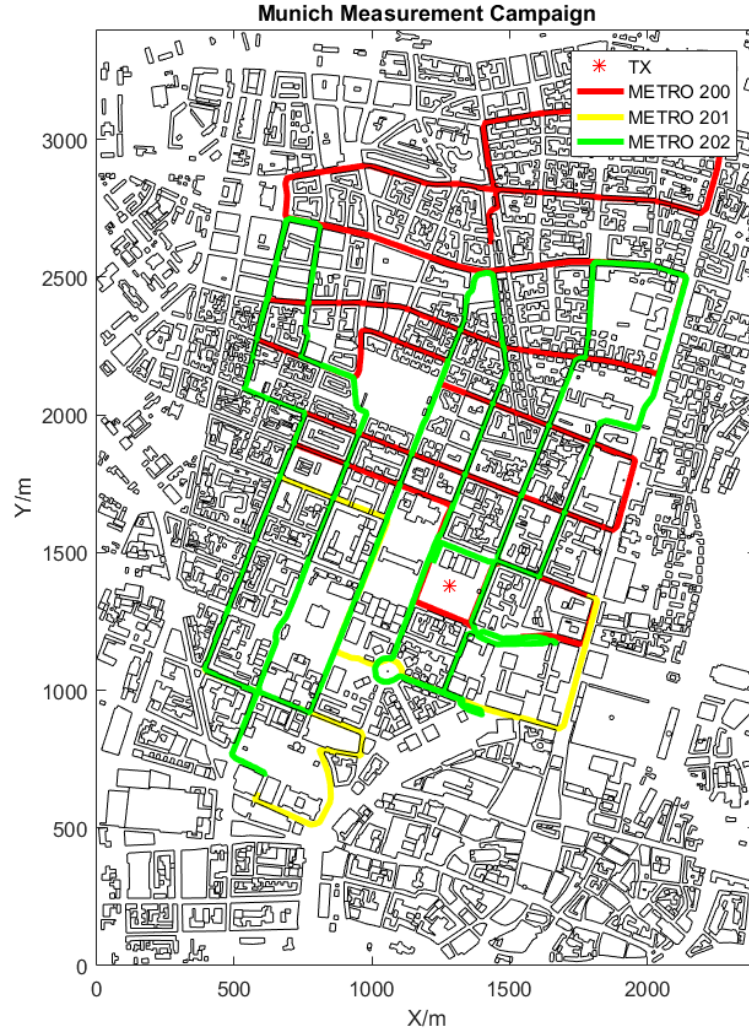


Figure 3-15: Measurement campaigns in Munich downtown area with three routes, ‘METRO 200’, ‘METRO 201’ and ‘METRO 202’.



In order to validate our 2.5D ray launching model, we ran a full simulation of the Munich downtown area where channel sounding measurements at 947MHz along three different routes are accessible. This data has been created during the COST 231 action as described by Damosso (1999) and is now publicly available at Mannesmann Mobilfunk GmbH, Germany [36, 46]. Figure 3-15 shows the actual layout of the area as well as the emitter location and three routes shown in different colours. Further analysis of this scenario can be found in Chapter 5.

The scale of this scenario is  $2.4\text{km} \times 3.4\text{km}$  and contains 2088 buildings, i.e., 17455 walls. Also within this area are 1758 trees accessible from the Overpass Turbo database. The emitter was located at (1281.36m, 1381.27m) and 13m above the ground, while the receiver was being held at a height of 1.5m in all three cases.

Discrete measurements of path loss were made along these routes with nearly the same distance interval. Omni-directional antennas were used during the measurements. A total of 970, 355 and 1031 observations were recorded along the three routes, respectively. The trends of path loss variation against sampling sequences are shown in Figure 3-16.

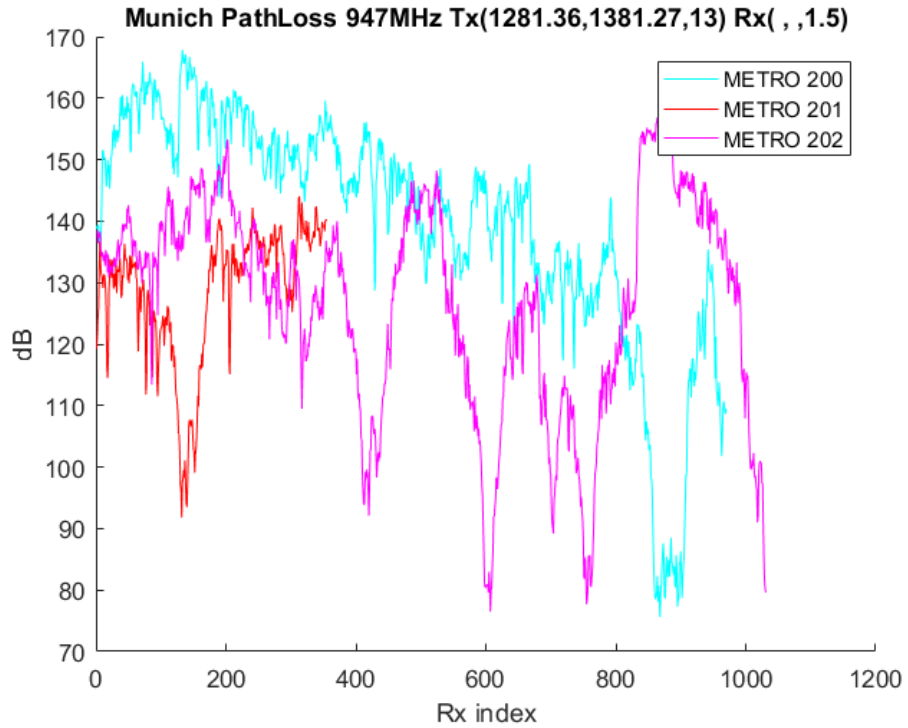


Figure 3-16: Pathloss trends along the three measurement routes on the Munich scenario.

For the simulation, the Origin point during measurement was found and aligned

with that in the model. A transmitted power equalled to 1 W with omni-directional radiation pattern was established. After setting up buildings and trees, an average building height of 18.7m was adopted for roof-top diffraction loss analysis. Due to incomplete data on building materials, we assumed they were all concrete regardless of thickness. Reflection and diffraction limits were set as seven and one, respectively. Three orders of wall penetration and one order of indoor reflection were considered. Reception spheres of 5 m radius which inscribes the grid of pixels was allocated across the entire scenario. The RSS for each sampling position was taken from the nearest reception sphere. As a result, we have connected those discrete RSSs into data series to obtain path losses. The path losses are plotted against the original measured data along the three routes as shown in Figures 3-17, 3-18 and 3-19.

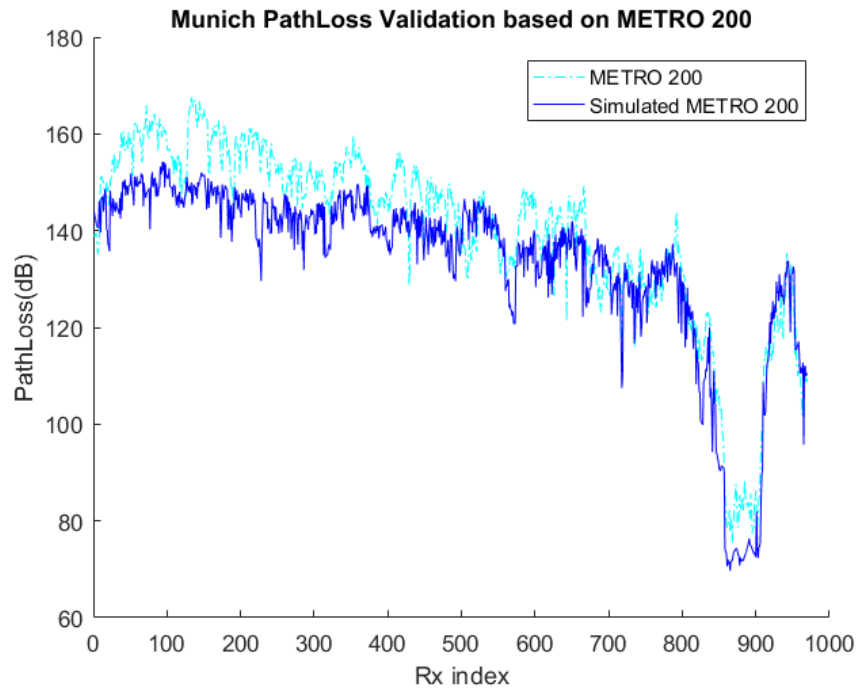


Figure 3-17: Simulated RSSs along ‘METRO 200’ compared to measurements.

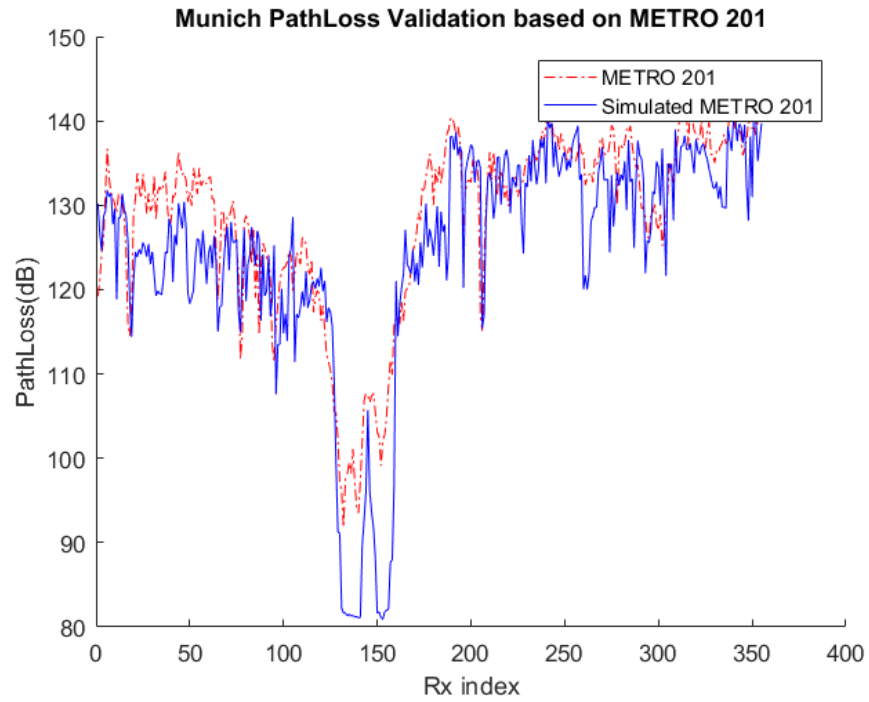


Figure 3-18: Simulated RSSs along ‘METRO 201’ compared to measurements.

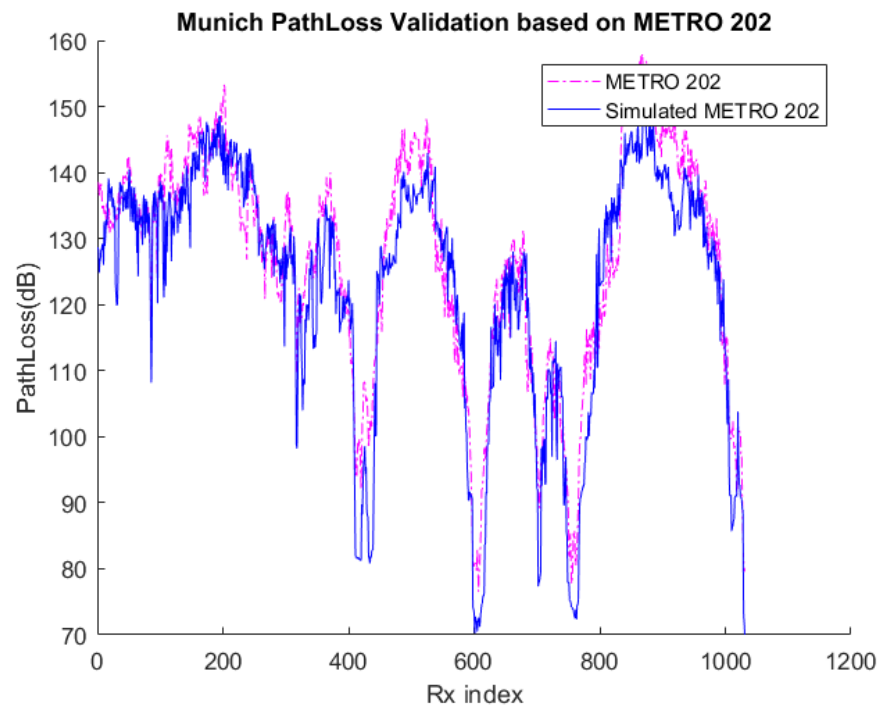


Figure 3-19: Simulated RSSs along ‘METRO 202’ compared to measurements.

Generally, the simulated path loss variations show a remarkable agreement with the measurements. The major trends of fluctuations which are presumably due to the mobile device threading through LOS and NLOS areas have been successfully depicted. The path loss predications show a good agreement with the measured data. This is true for both the close LOS areas as well as for the more distant NLOS areas where roof-top diffraction is important.

Table 3.2 shows the Mean Absolute Errors and Standard Deviations between simulation results and measurements along the three routes. As can be seen, the Standard Deviation always remains less than 7dB along any route. The Mean Absolute Error also shows good agreement. Overall, the ray launching model is shown to be effective for a relatively large and complicated urban scenario.

Table 3.2: Mean Absolute Errors and Standard Deviations along the three routes in the Munich scenario

<b>Route</b>	‘METRO 200’	‘METRO 201’	‘METRO 202’
Mean Absolute Error(dB)	7.7373	5.7189	6.4210
Standard Deviation(dB)	6.6026	6.4883	6.8691

Nevertheless, disagreements can be found. Several factors may potentially account for these errors. Firstly, the environmental entry does not provide complete details, e.g., the precise building geometries, heights, and materials, which will certainly restrict the accuracy. Secondly, the 2.5D model has its limitations in dealing with certain propagation primitives, such as ground reflections, or higher orders of diffraction involving both horizontal and vertical edges. Thirdly, the presence of vehicles and pedestrians are very much likely to affect the propagation circumstances in practice.

Taking the longest measurement route ‘METRO 200’ for example, we plotted discrete simulated path losses against measurements as shown in Figure 3-20. This scatter plot leaves out the spatial consecutiveness of measurements while regarding each path loss value as an individual event. The linear correlation coefficient along ‘METRO 202’ equals to 0.9089 which demonstrates good agreement. Generally, the simulations and measurements are linearly dependent which verifies the effectiveness of our ray launching model. However, we do notice that in the smaller path loss range the simulator tends to underestimate the actual measured loss. This is most likely due to limitations in incorporating all environmental information, e.g., traffic and pedestrians which prevent LOS propagation. The routes ‘METRO 200’ and ‘METRO 201’ produce correlation coefficients of 0.9236 and 0.8725, respectively. Thus, all routes show good agreement between simulation and measurement.

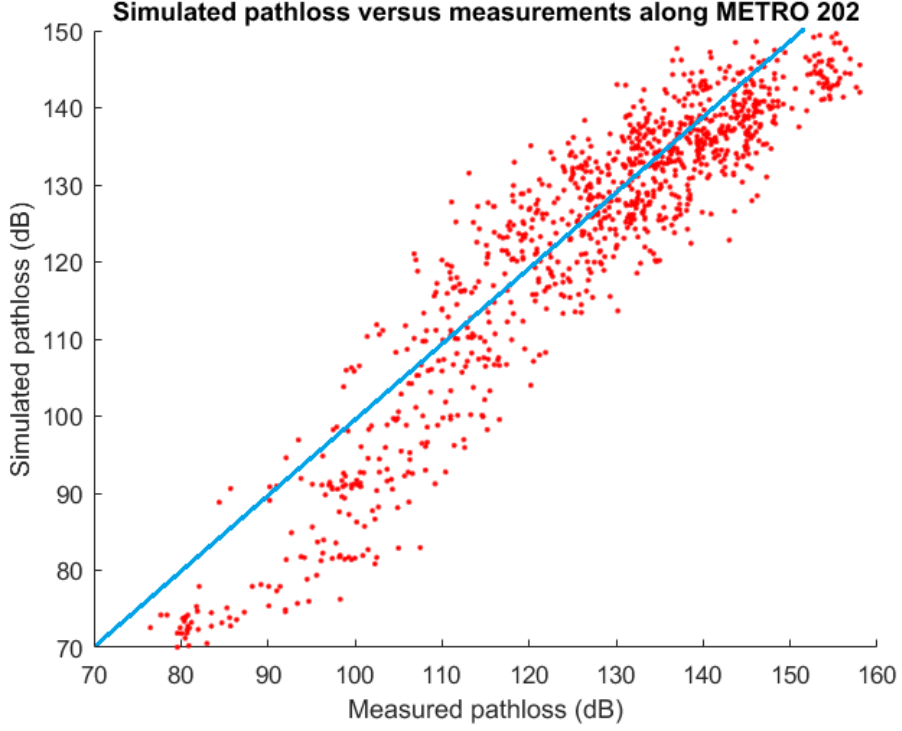


Figure 3-20: Scatter plot between simulated pathloss and measured pathloss along ‘METRO202’. The solid blue line shows the trajectory of correlation coefficient equalled to unity.

Wave propagation modelling aims at producing as accurate channel estimations as possible. However, even the best current commercial ray tracers are not able to achieve perfection. The core of this research lies in the leveraging of coarse but efficient channel predictions to accomplish navigation and source localization. These are to be discussed in the following chapters.

### 3.6 Summary

This chapter expands the design procedures of the 2.5D ray launching model. This model integrates geometric information of the environment along with the most significant propagation mechanisms of free-space propagation model. Four principal propagation mechanisms are incorporated, including reflection, diffraction, wall penetration and roof-top diffraction. A virtual reception sphere is used to detect arriving rays. This model has been validated against measurement campaign in Munich city center demonstrating very good agreement.

## Chapter 4

# Navigation

Many modern navigation systems are heavily reliant upon the GNSS. Whereas, densely built urban environments which are vulnerable to multipath propagations undermine the precision and reliability. There is currently growing interest in localization based on signals of opportunities in which propagation parameters observed from ground emitters may allow urban navigation. Although these methods take advantage of the multipath, they usually require laborious collections of ‘truth data’ at various locations. Furthermore the temporal stability of the environment often reduces the effectiveness.

In this chapter, we propose to obtain channel parameters from modelling instead of field measurement, and exploit a fingerprinting approach to bond those parameters to locations. At the heart of this method lies the definition of the ‘location fingerprint’ and the development of a reliable localization algorithm. We demonstrate what components are essential in constituting the fingerprint and how to extract them from our 2.5D ray launching model. It is also found that Artificial Neural Networks are a valid tool to generate the mapping functions. In pursuit of an optimized mapping function, we investigate a series of variables and associated errors affecting the on-line navigation process. Overall, the proposed navigation system shows very good accuracy and reliability.

### 4.1 Location fingerprint

Much research in the literature has validated the feasibility of using field measurements to locate a mobile device. The most popular primitive to exploit is RSS because the embedded device is easy to install and there are plenty of wireless communication access points. RSS measurements are used for indoor localization systems in [118, 132].

Abid *et al.* [4] experimented and validated using RSS measurements to navigate in urban areas. Wilfinger *et al.* [127] proposed a position determination algorithm based on RSS readings from passive RFID (Radio Frequency Identification Devices) tags. The digital television, radio cellular base station, Wi-Fi source etc. are common places in urban environments nowadays. As a result, signals of opportunities are rich resources to take advantage of.

Although conventional triangulation-based localization methods rely very much upon TDOA and AOA measurements, their reliabilities drop significantly in severe multipath environments [101]. Despite some disadvantages, the capability of using TDOA and AOA to navigate has recently drawn considerable research attention [22, 110] as long as assuming LOS-only propagation.

In the literature review, we have addressed that spatially varying feature is pervasively seen in the RSS, TDOA and AOA distributions of these signals. Therefore, the core concept of our navigation system is utilizing the spatially-distributed channel characteristics to discriminate individual physical locations. Even though the multipath breaks the elegance of simple dependence in terms of power attenuation, delays on travelled distance and angular spread, it brings about diversity and irregularity into the data distribution which gives rise to viability of location discrimination.

An advantage of propagation modelling is that it generates channel response estimations on top of a balance between accuracy and rapidity which can be balanced. Given a description of the environment, we are able to predict receiver observations rather than carrying out laborious field measurements. Even if a detailed geometry is to be simulated, it is likely to take much less than measurement. Moreover, changes in the environment, e.g., construction of a new building, adding interfering emitters and the presence of traffic, are merely costing another round of simulation. Whereas, such changes could lead all measurement efforts to be in vain.

An essential step before establishing a navigation system is to extract geographical features from channel characteristics. By way of example, a virtual enclosed box is built in the last chapter. The same geometry is used with a receiver located in a NLOS area as is shown in Figure 4-1. Subsequently, the PDP and Angular Spread are obtained and displayed in Figure 4-2 and Figure 4-3. It can be seen that the features extracted from this location are significantly different from those in Figures 3-13 and 3-14. Our hypothesis is that the characteristics are hopefully distinctive in comparison to any other receiver location. To solve the navigation problem, we need to identify a method of mapping the received characteristics to a location. A popular technique, known as fingerprinting, is adopted here to solve this data mining issue.

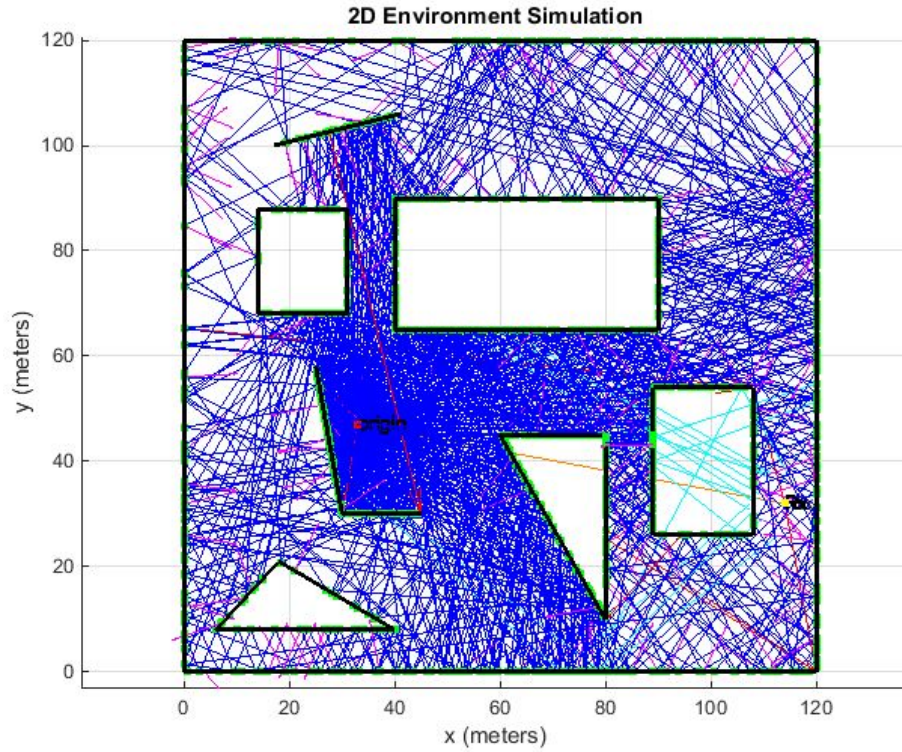


Figure 4-1: Propagation paths to a NLOS receiver at (114m, 32m) on the virtual seven-object scenario. All reflected rays are presented in blue, whereas, only penetrated, diffracted and roof-top diffracted rays that are received are shown in light blue, red and orange, respectively.

In this thesis, a fingerprint which represents a physical location is termed as location fingerprint. Ideally, a location fingerprint must be unique. Taking received power for example, an omni-directional antenna may find similar readings in completely opposite directions to the emitter. Thus, features of limited dimensions alone usually cannot make an effective location fingerprint. Previous fingerprinting approaches, e.g., [113], that absolutely rely on RSS measurements do not produce good accuracy performances with less than 20 signals of opportunities. Although more components in a location fingerprint usually guarantees better uniqueness, *a-priori* knowledge of many sources becomes critical. In this research we aim to reducing this number to less than ten by incorporating TDOAs and AOAs as well as enhancing the localization algorithm.



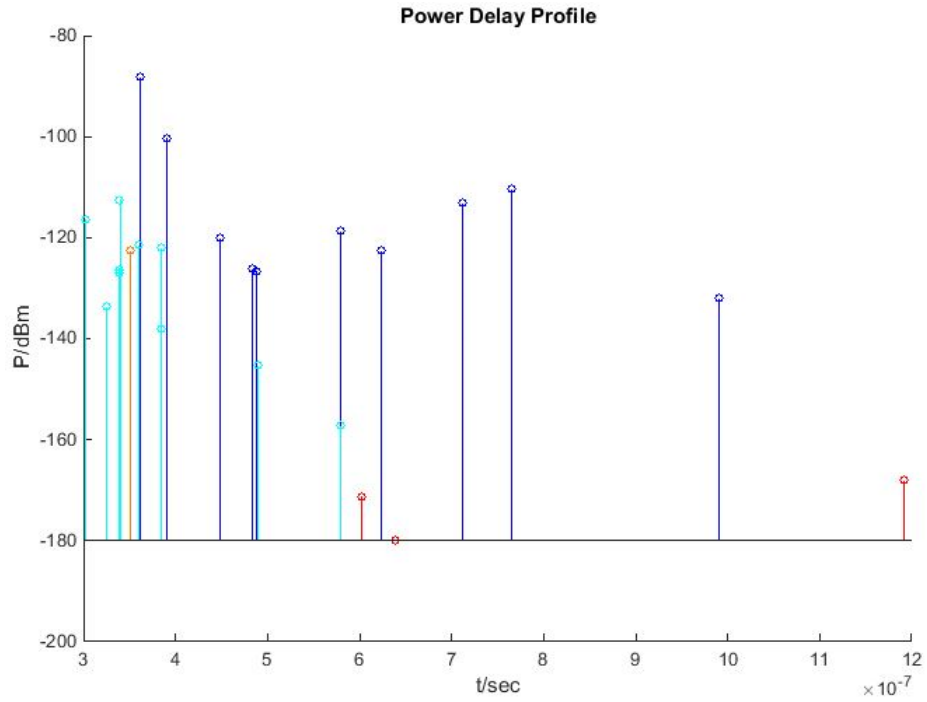


Figure 4-2: PDP of the NLOS receiver at (114m, 32m).

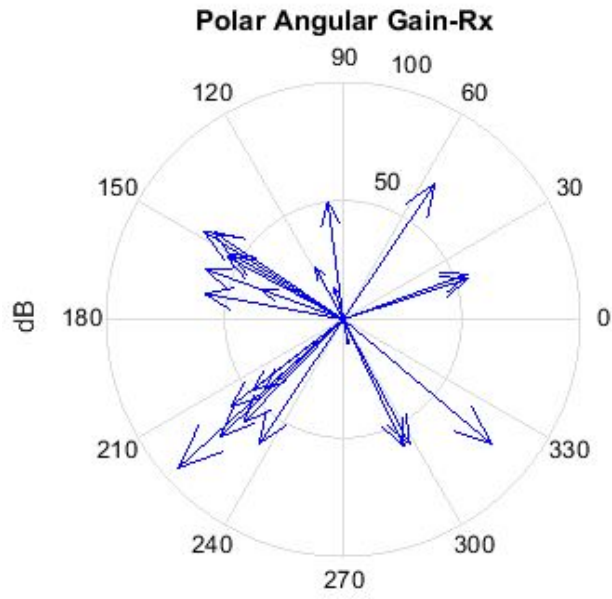


Figure 4-3: Angular Spread of the NLOS receiver at (114m, 32m).

A location fingerprint primitive which is the element of a location fingerprint should be repeatable. This means that different users should expect a similar reading using

an identical device at the same location. Only when this expectation holds could a localization algorithm work. Provided good accuracy capability from the ray launching model as well as easily measurable electromagnetic characteristics, it ought to be possible to create a suitable location fingerprint primitive [140]. These primitives must be relatively stable in an ever changing electromagnetic environment (e.g., due to pedestrians and vehicles), such that training and deep learning result in reliable and resilient navigation solutions.

From the PDP and Angular Spread, a series of eigenvectors with fingerprinting significance can be extracted, such as RSS, MED (Mean Excess delay), RMSDS (Root-Mean-Square Delay Spread) and AOA (Angle of Arrival). These eigenvectors all possess a degree of uniqueness and repeatability which form good location fingerprints.

Given the received power,  $P_r$ , of each arriving ray, the average received signal strength,  $RSS$ , at location  $(x, y)$  can be approximated as Eqn. (4.1);

$$RSS_{(x,y)} = \sqrt{\sum_{i=1}^N P_{r_i}^2} \quad (4.1)$$

where  $N$  is the total number of impulses. The  $MED$  at  $(x, y)$  is given by;

$$MED_{(x,y)} = \frac{\sum_{i=1}^N P_{r_i} t_i}{\sum_{i=1}^N P_{r_i}} \quad (4.2)$$

where  $t_i$  is the absolute time delay of each arriving ray. The  $RMSDS$  at  $(x, y)$  can be expressed as;

$$RMSDS_{(x,y)} = \sqrt{\overline{\tau^2} + MED_{x,y}^2} \quad (4.3)$$

where  $\overline{\tau^2}$  is derived from;

$$\overline{\tau^2} = \frac{\sum_{i=1}^N P_{r_i} t_i^2}{\sum_{i=1}^N P_{r_i}} \quad (4.4)$$

In this thesis, we assume only the most significant AOAs can be measured and only with low accuracy. Hence, we suppose the first AOA is exactly the direction in which the strongest received power is detected. The second AOA refers to the direction of second strongest ray and so on. A direction is a vector which has to be presented with at least two scalars. Thus, we usually represent the AOA as Cartesian projections;

$$\overrightarrow{AOA} = (\cos \overrightarrow{AOA}, \sin \overrightarrow{AOA}) = (AOA1X, AOA1Y)$$

where AOA1X refers to the  $x$ -axis Cartesian projection of the most significant arrival,

and AOA1Y refers to the  $y$ -axis Cartesian projection of the most significant arrival.

The projected scalars of an AOA are always between  $[-1, 1]$ . Similarly, we would like to normalize the RSS, MED and RMSDS within an equivalent range depending on the measurement requirements. For example, RSSs are truncated between  $[-140\text{dBm}, -60\text{dBm}]$  because below  $-140\text{dBm}$  signals are hardly detectable and signals in the far field are generally below  $-60\text{dB}$ .

Ultimately, we aim to integrate these features by merging the normalized values together to construct a location fingerprint. Given  $M$  signals of opportunities, a location fingerprint as a function of 2D coordinates can be written as

$$\mathbf{Fingerprint}(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} \text{RSS}_1 & \text{MED}_1 & \text{RMSDS}_1 & \text{AOA1X}_1 & \text{AOA1Y}_1 \\ \text{RSS}_2 & \text{MED}_2 & \text{RMSDS}_2 & \text{AOA1X}_2 & \text{AOA1Y}_2 \\ \text{RSS}_3 & \text{MED}_3 & \text{RMSDS}_3 & \text{AOA1X}_3 & \text{AOA1Y}_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \text{RSS}_M & \text{MED}_M & \text{RMSDS}_M & \text{AOA1X}_M & \text{AOA1Y}_M \end{bmatrix}$$

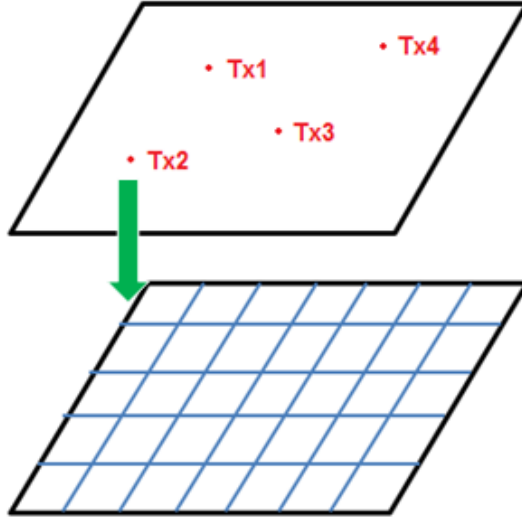


Figure 4-4: A schematic diagram on fingerprints generated by opportunistic sources.

In the physical layer, each known source of opportunity would generate a row of fingerprint primitives regarding each location as illustrated in Figure 4-4. Each column stands for an extracted and normalized electromagnetic feature for a location. A loca-

tion can be redefined by tagging such fingerprint matrix, and we would like to develop a localization algorithm using these tags.

## 4.2 Localization algorithm

The location fingerprint is obtained to represent the identity of each geometric location which incorporates RSS, MED, RMSDS and AOA. Then the fingerprint can be further expanded by considering contributions of opportunistic signals, e.g., digital television, radio broadcasts, cellular communications, Wi-Fi signals etc. In this way, the dimension of the location fingerprint is magnified by the number of opportunistic sources in the scenario. Since the source locations are usually different in practice, their propagation geometries differ from each other. On the other hand, as the propagation coefficients such as the reflectivity of surfaces are frequency dependent, the receiving eigenvectors will vary to allow diversity for the fingerprints. Overall, the more signals available in the scenario the richer each fingerprint will be. Modern urban areas generally satisfy this condition as a variety of transmitters are sited for different telecommunications purposes.

The localisation algorithm is developed based on the idea of exhaustive searching and matching of the location fingerprints. Given the known locations of sources in the area of interest, we run the simulation for all candidate locations of the receiver. These location fingerprints are subsequently stored in a database. Suppose a mobile user wanders through the given scenario along an unknown path while continuously reporting their measured fingerprints. The system will find the best match of location fingerprint within the database to decide their location. Note that for the same frequency we assume there is only one source such that the sources do not interfere.

### 4.2.1 Mapping function

A fingerprint-based localization algorithm is generally comprised of on-line phase and off-line phase. During the on-line phase, when an entry of observed fingerprint is given the algorithm is expected to return a best matched location among the fingerprint database. Provided that we have successfully defined and extracted fingerprints in the off-line phase, a reliable mapping function bridging the fingerprints and the locations becomes the key.

Apparently, the location in a 2D coordinate system can be expressed as  $(x, y)$  of which the absolute values are related to the origin. On the other hand, the fingerprints have been defined as matrix comprised of electromagnetic features including RSS, MED, RMSDS and AOA. We attempted to visualize the fingerprint primitives by

allocating a grid of receivers of  $1\text{m}\times 1\text{m}$  square and plotting these primitives against the coordinates. As can be seen from Figure 4-5, a heat map of RSS is built for the scenario where brighter coloured pixels imply higher RSS and darker pixels indicate lower RSS detected. Not only are the shading effects of objects presented, but also the radiation pattern contours of the emitter in the horizontal domain are also reflected as provided in Figure 3-2.

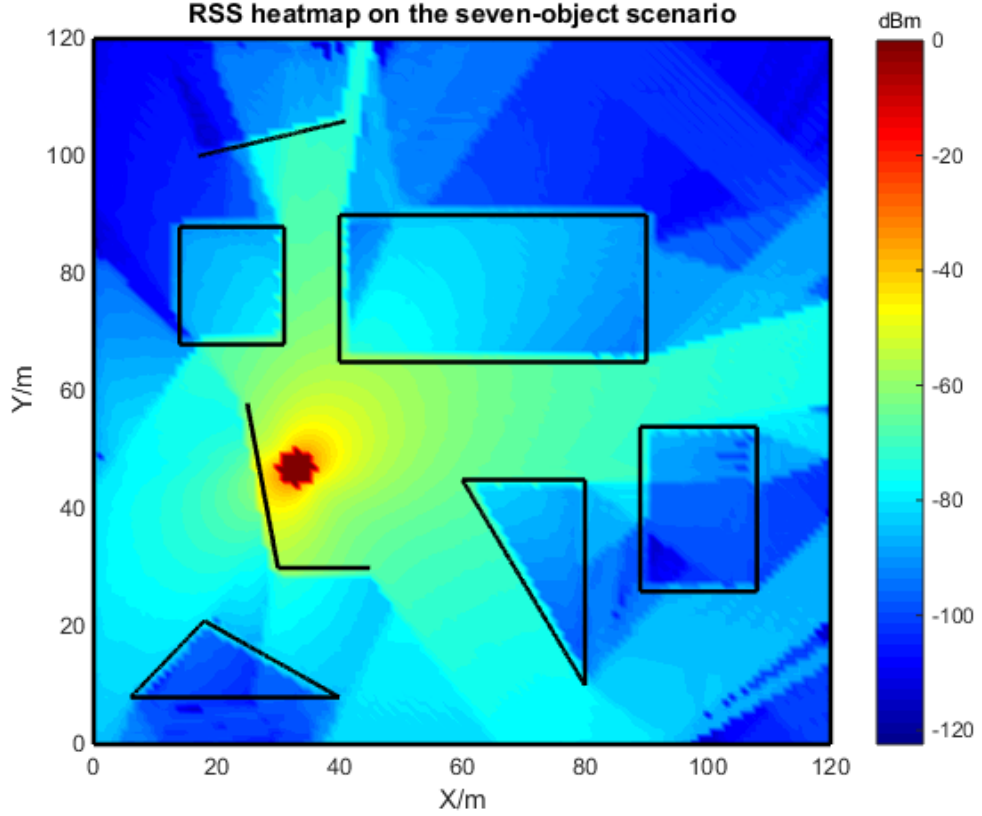


Figure 4-5: RSS heat map in the enclosed box scenario with seven objects given an emitter transmitting 0dBm at (33m, 47m).

A significant characteristic of the RSS data distribution is their discontinuous nature. Although the pixels turn darker as they move away from the emitter in most circumstances, the geometry of the objects brings about different degrees of shading and illumination which are highly stochastic. This means linear regression of the spatial distribution of the location fingerprints is almost impossible.

Due to the fact that the simulation results heavily depend on the environmental settings which usually vary in a range of uncertainty, a location fingerprint has to be tolerant to errors. In order to reliably map the location to the fingerprint, the

fingerprints must be distinct for every location even in the presence of errors.

Moreover, a preferable mapping function should be fast enough to provide real-time localization. In order to deal with the discontinuity of high-dimensional fingerprints, Artificial Neural Network is used to generate accurate, fast and dynamic mapping functions.

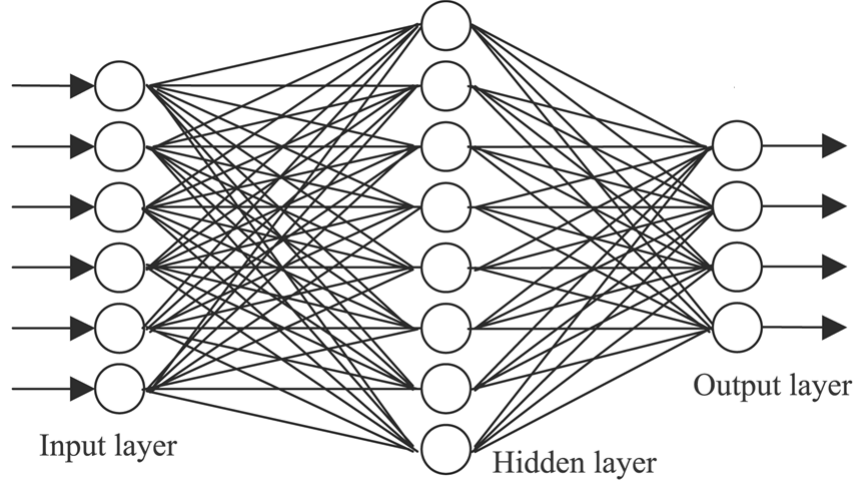


Figure 4-6: A schematic diagram of a one-hidden-layer Artificial Neural Network.

Reviewing the literature, we have concluded that feed-forward back-propagation Neural Network is the most appropriate in dealing with discontinuous data entries [18, 99]. As shown in Figure 4-6, it typically consists of three layers: the input layer; the hidden layer (one or above); the output layer. Data to be trained are fed into the input layer. Each neuron in the hidden layer stores an unlabelled characteristic. The output layer directly represents the expected training target. All neurons between any two layers are fully connected. During training, for each connection a series of variables are constantly being optimized according to the feedback from the output difference.

#### 4.2.2 Training

To demonstrate the effectiveness of using the Neural Network to generate a mapping function, we have investigated a typical residential area in the Hoxton district of London. The area of interest is 700m×550m. Since it is relatively large, we compromise the resolution slightly to get a reasonable computation time by assigning the pixel size as 5m×5m. In total, there are 15400 receiver pixels. Some 120 objects with 1376 walls and 123 trees are downloaded from the OpenStreetMap. Without precise knowledge on the sources in this region, we assumed six radio access points of different

narrow band frequencies as listed in Table 4.1.

Table 4.1: Virtual emitters allocated to the Hoxton scenario

Source	Frequency(GHz)	Power(dBm)	Location(m)	Height(m)
FM Radio broadcast	0.088	30	(490, 150)	30
DAB radio signal	0.225	40	(110, 170)	20
GSM down link	0.9	10	(50, 500)	10
GSM upper link	1.8	10	(550, 480)	10
Wi-Fi 2.45 GHz	2.45	20	(315, 340)	2.5
Wi-Fi 6 GHz	6	20	(330, 350)	1.5

These sources of different powers are dispersed across the map with the aim of producing diversity in the fingerprint data distribution. The receiver is always assumed to be 1.5m above the ground. A sketch of the geometry and Wi-Fi location is shown in Figure 4-7. Due to the absence of building heights, we assume every roof-top is 15 m above ground.



Figure 4-7: Hoxton district layout with a virtual Wi-Fi transmitter marked as a red dot at (315 m, 340 m).

We have subsequently simulated the receiver observations using the ray launching

model and built a location fingerprint database. The next step is training this database using the Neural Network. The target outputs of the training are the 15400 pairs of  $(x, y)$  coordinates. The remaining question is how we configure the inputs and Neural Network parameters.

The RSS plays an important part in categorizing the scenario into hierarchies of illumination. A 2D view of the RSSs at 2.45 GHz is shown in Figure 4-8. Due to diffraction and wall penetration losses there is significantly less signal strength detected in shadowed areas than LOS areas. In the LOS areas, the RSS keeps fading farther away from the transmitter as the free-space propagation loss increases. As can be seen in Figure 4-8, the multipath and shadowing effect results in remarkable discontinuity of the RSS distribution which helps to differentiate zones separated by walls. Nevertheless, the outwardly attenuating envelope of RSS still reveals how far away the receiver location is from the source.

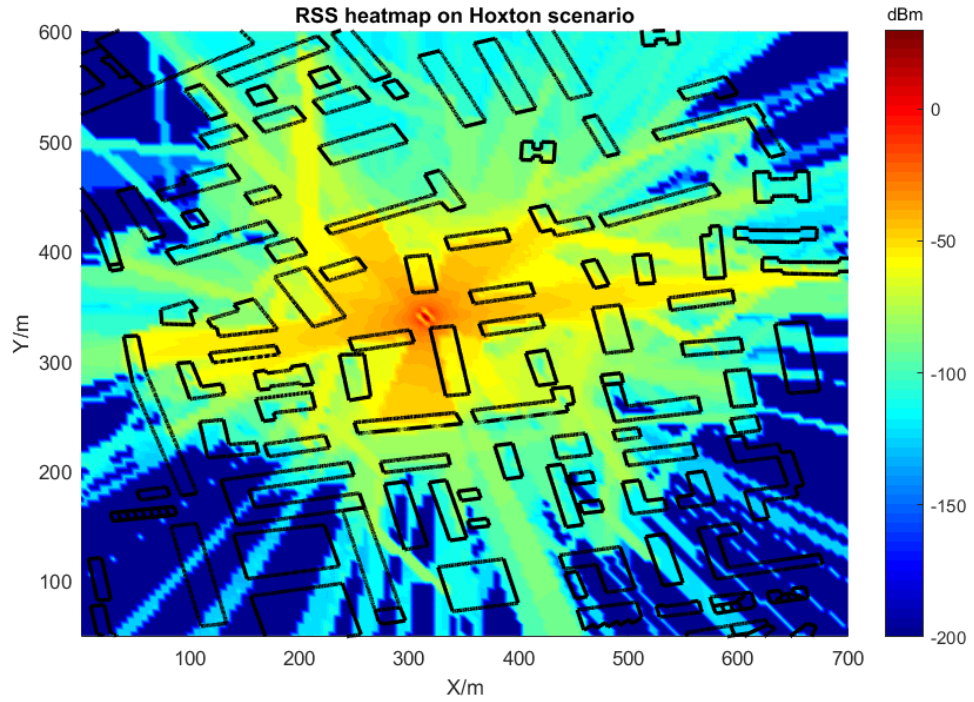


Figure 4-8: Simulated RSS heat map for the Hoxton scenario provided by a 20 dBm emitter of 2.45 GHz at (315 m, 340 m).

The MED and RMSDS are combined to depict the time delay of arrival as well as the arrival time clustering along the time axis. These properties reflect subtle differences in which arrivals make up the PDP. They are also expected to help discriminate receiver locations.



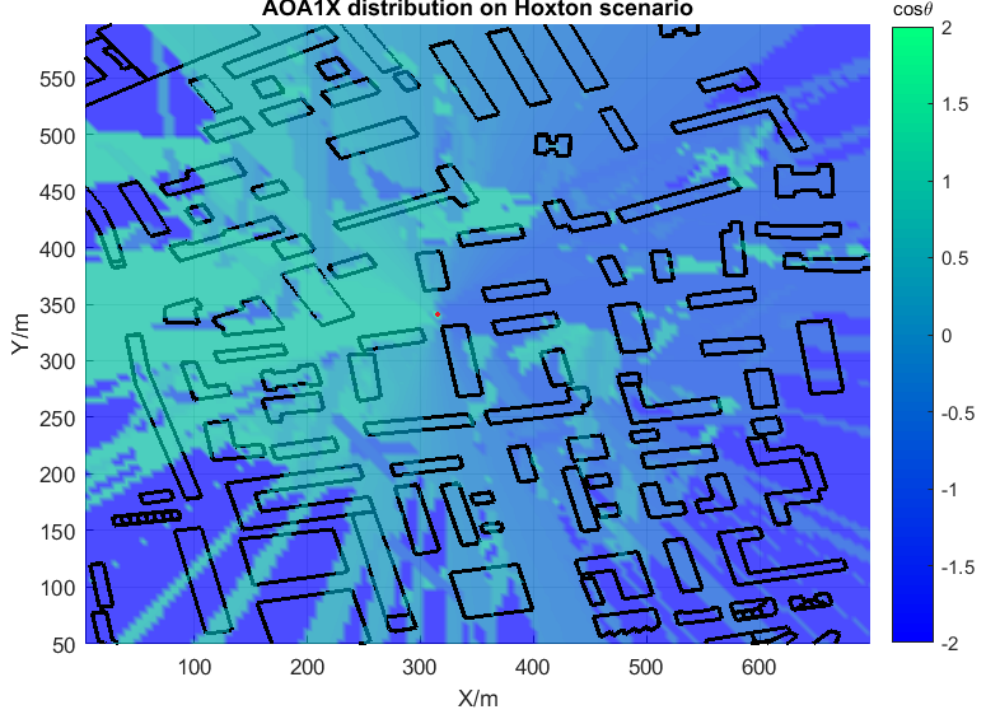


Figure 4-9: The most significant AOA x-axis Cartesian projection on the Hoxton scenario, where null pixels are assigned -2; transmitter location is assigned 2 marked as a red dot.

In LOS propagation, the AOA provides straightforward information on which direction the receiver is to the source, as can be seen in Figures 4-9 and 4-10. In addition to the arrival angle of the strongest signal, we are also interested in exploiting the 2nd and 3rd most significant arrival angles. Since the reception detection guarantees one ray received at a time, rays other than most significant arrival provide information on higher orders of illumination situations. For instance, two different receiver locations might have identical LOS arriving angles, whereas, their 2nd or 3rd strongest arrivals coming from diverse propagation paths offer evidence to distinguish them. Consequently, using the top three significant AOAs could not only divide the illuminated region into ‘pizza-slice’ like zones to localize, but help discriminate receiver locations on a smaller scale.

As a result, the number of columns for a location fingerprint increases to nine as each AOA contains two scalars. Hence, the row vector of the fingerprint matrix can be written as

$$[\text{RSS}_i, \text{MED}_i, \text{RMSDS}_i, \text{AOA1X}_i, \text{AOA1Y}_i, \text{AOA2X}_i, \text{AOA2Y}_i, \text{AOA3X}_i, \text{AOA3Y}_i]$$

Given  $M = 6$  opportunistic sources, a location fingerprint contains  $6 \times 9 = 54$  elements. Hence, the input layer to the ANN is comprised of 54 entry neurons.

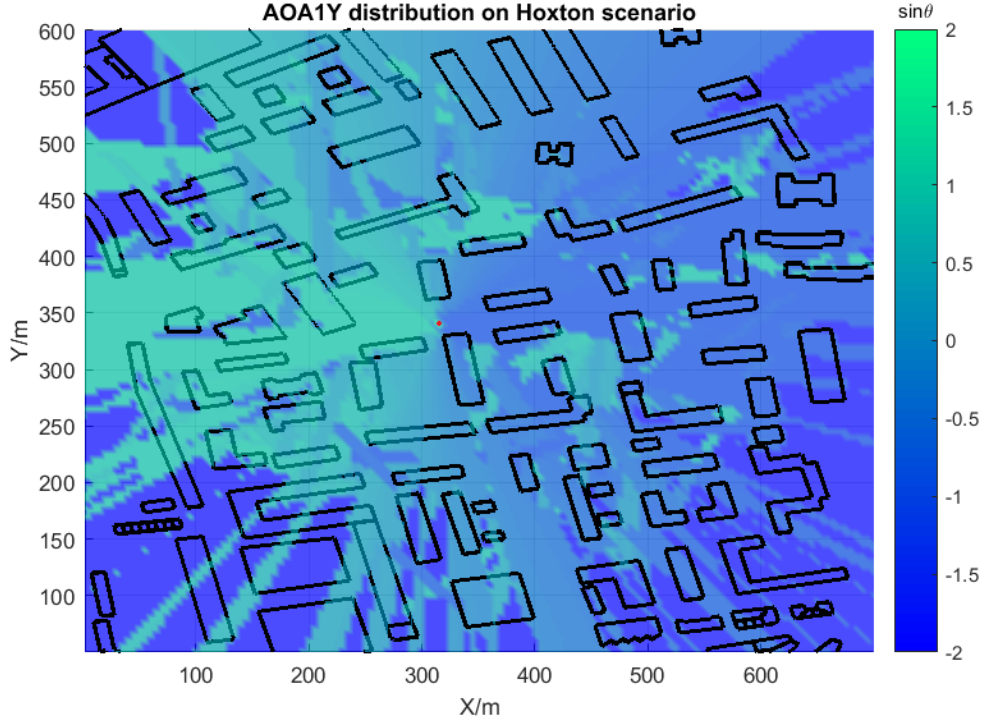


Figure 4-10: The most significant AOA y-axis Cartesian projection on the Hoxton scenario, where null pixels are assigned -2; transmitter location is assigned 2 marked as a red dot.

The Matlab ANN toolbox provides a comprehensive training platform. In addition to the input and output layers, a series of variables need to be specified. In a feed-forward back-propagation network the signals flow in one direction from input, via the hidden layers, to target. Whereas, the error which is obtained from subtracting target from the current training output constantly back-propagates to the input to reduce the error for the next iteration. Network type of feed-forward back-propagation is often cited to produce the fastest convergence and is outstanding in pattern recognition.

The ‘TRAINLM’, short for Levenberg-Marquardt training algorithm, is acknowledged as one of the most accurate and fastest converging functions [115]. It usually outperforms other training functions in both curve-fitting and pattern recognition problems. The downside of ‘TRAINLM’ is a larger memory space requirement which also results in longer execution time. Other popular training functions include ‘TRAINRP’ (Resilient backpropagation algorithm), ‘TRAINSCG’ (Scaled Conjugate Gradient backpropagation algorithm), ‘TRAINBFG’ (BFGS quasi-Newton backpropa-

gation algorithm). We have examined their performance which either exhibited limited accuracy or lacked stability in achieving a good result in comparison to ‘TRAINLM’.

The adaption learning function, ‘LEARNNGDM’ (Gradient Descent with Momentum weight and bias learning function), is recommended for the incremental training [1]. ‘MSE’, short for Mean Square Error, is used to measure the performance. Detailed network configuration properties are captured in Figures 4-11 and 4-12.

**Network Properties**

Network Type: Feed-forward backprop

Input data: (Select an Input)

Target data: (Select a Target)

Training function: TRAINLM

Adaption learning function: LEARNNGDM

Performance function: MSE

Number of layers: 3

Properties for: Layer 1

Number of neurons: 48

Transfer Function: LOGSIG

View Restore Defaults

Figure 4-11: Neural Network user interface panel.

Properties for: Layer 2

Number of neurons: 32

Transfer Function: TANSIG

Figure 4-12: Neural Network settings.

According to the literature on ANN [124], multilayer Neural Networks are deemed more capable of handling discontinuous entries. As a compromise between speed and accuracy, two hidden layers and one output layout are assigned. Demuth [26] suggest that over half the neurons of the last layer should be allocated to a hidden layer to achieve better convergence performance. Following the approach of [26], in the Hoxton example, we assigned 48 neurons to the first layer and 32 to the second. The sum of the weighted inputs and the bias from the last layer forms the input to the transfer

function which generates the output for each neuron. Note that the selection of transfer function for each layer effects how the weights and biases proceed to reduce the error in every iteration. Table 4.2 lists the characteristics of common transfer functions.

Table 4.2: Characteristics of transfer functions in the Neural Network

Name	Expression	Output range	Recommendation for use
<i>Log-sigmoid</i>	$a = \text{logsig}(n)$	$(0, 1)$	Multilayer networks
<i>Tan-sigmoid</i>	$a = \text{tansig}(n)$	$(-1, 1)$	Pattern recognition problems
<i>Linear</i>	$a = \text{purelin}(n)$	$(-\infty, \infty)$	Function fitting problems

Based on trial and error, we found that using *Log-sigmoid* for the first layer and *Tan-sigmoid* for the second yielded the best convergence performance. A schematic diagram of the configured ANN for the Hoxton example is shown in Figure 4-13.

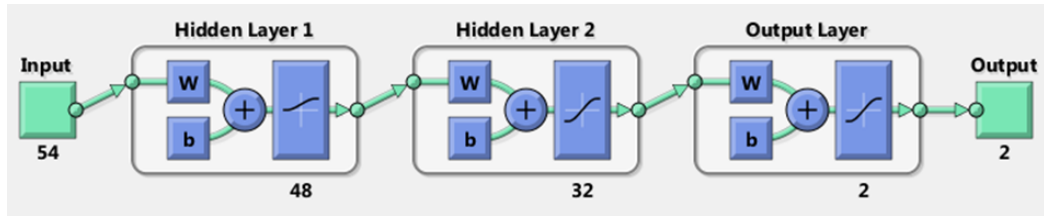


Figure 4-13: Neural Network configuration diagram.

Using this configured ANN, we are able to train the input fingerprints for 15400 iterations, equivalent to the total number of receiver candidates, with respect to the 2D coordinates of the required locations. The training result panel is displayed in 4-14. It is worth noticing that  $Mu$  represents the training gain of the outcome error compared to the initial error. This particular training process took six iterations and was performed within two minutes. The mean square error converged to  $169\text{m}^2$  which implies a mean mapping error of 13m. Considering the fact that a single receiver pixel takes up a  $5\text{m} \times 5\text{m}$  square, the mean error is good.

It is definitely true that by setting a more critical training gain before stopping one could possibly obtain an even better training performance. However, an overly trained system may produce worse estimation in practice because the intermediate inputs may find erroneous outputs even though the 15400 samples are perfectly fitted. A closer look at the accuracy performance is given in the next section.

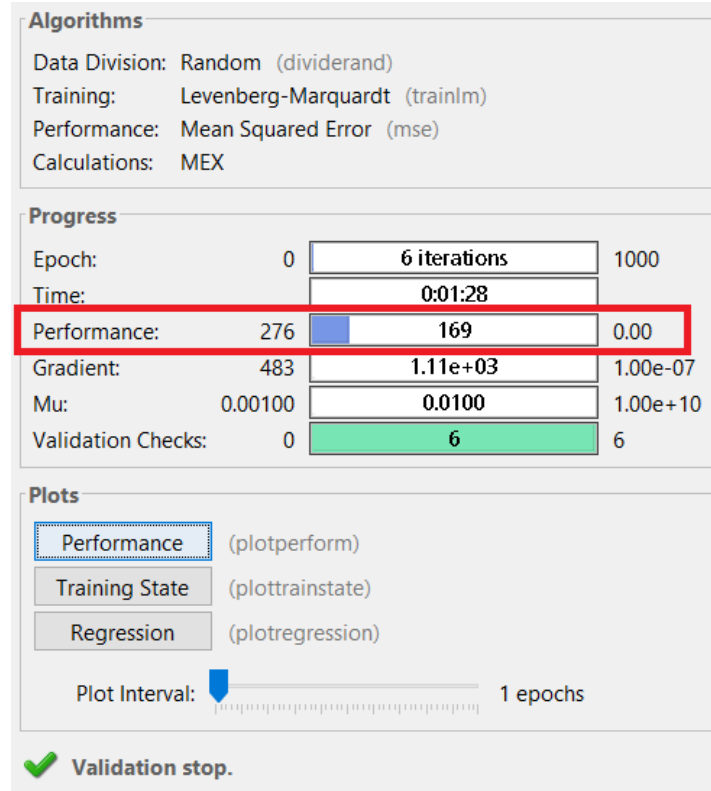


Figure 4-14: Training result panel for the Hoxton example.

### 4.3 Accuracy and reliability

Certainly, a theoretically minimal error is not the only goal of a localization algorithm. A variety of factors which might affect the training performance need to be studied, and it is questioned if this method is applicable to any urban scenario. Moreover, the mapping function has to be resilient to noise and deviations applied to the input since observed fingerprints are likely to differ from the simulation results which are used as training data. To find explanations for the above concerns, we carried out experiments by manipulating variables for the Hoxton district example and have drawn conclusions based on our observations.

A very straightforward means to improve the accuracy is to make the performance measure, Mean Square Error, as small as possible. This involves several manageable factors, such as the selection of training inputs among RSS, TOA and orders of AOA, the number of opportunistic sources, the resolution of allocated receiver pixels, the number of layers and neurons in each layer, the transfer functions, the expected training gain and so on.

### 4.3.1 Number of opportunistic sources

In the previous Hoxton example, we chose six opportunistic sources, specified in Table 4.1, as location fingerprint generators and gained 13m of mean error after training. Figure 4-15 shows the relationship between the mean distance errors and the number of sources available. Generally, as more opportunistic sources are incorporated the mean error decreases, whereas, the accuracy gain starts to drop when the number of sources is greater than six. Up until ten opportunistic sources, enriching the location fingerprint by incorporating more makes little contribution to the overall mapping performance. Therefore, it is a matter of balance between the number of sources and complexity of a location fingerprint to obtain a reasonable training performance.

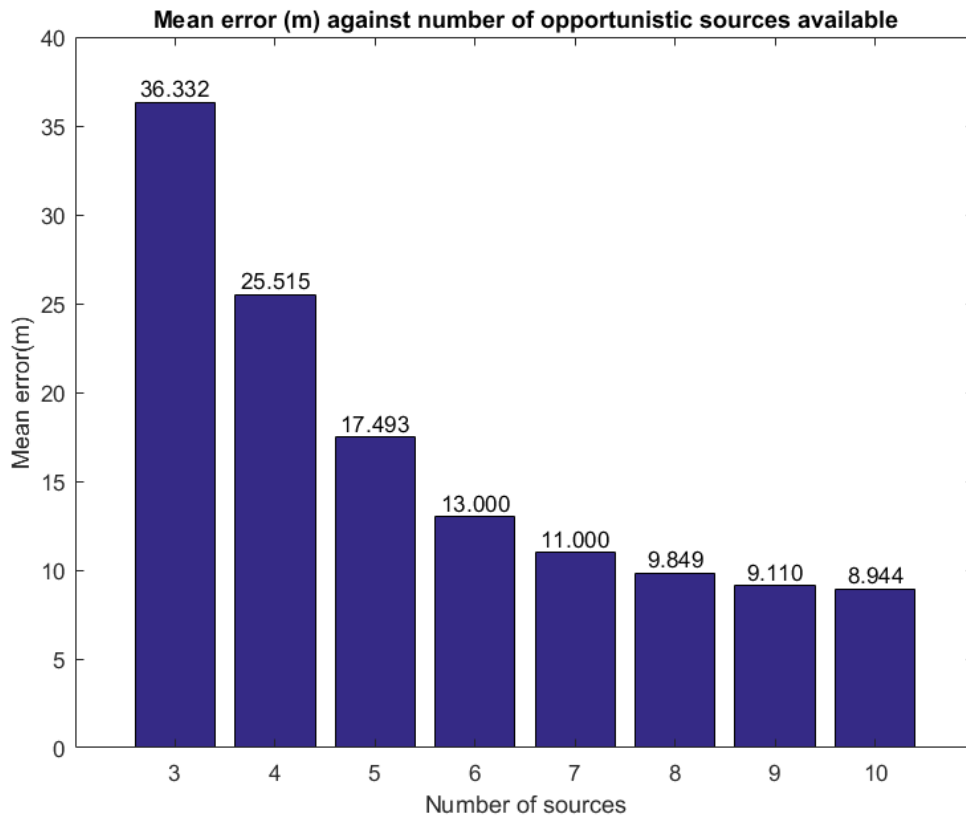


Figure 4-15: Mean error distance as a function of number of opportunistic sources available.

### 4.3.2 Location fingerprint components

We defined a location fingerprint should comprise RSS, MED, RMSDS and top three AOAs because their spatial distributions may be used to discriminate locations,

and that they are measurable. Nevertheless, the significance of each primitive requires further investigation. To measure the contributions of these primitives, we excluded one component at a time and trained the rest. Table 4.3 lists the exclusion conditions of fingerprint primitives and their corresponding mean errors. It can be seen that the exclusion of the most significant AOA has the most severe impact on performance, followed by the RSS. Time delays and higher orders of AOA are less crucial but still result in remarkable deterioration compared to the original 13m error.

Table 4.3: Mean errors against combinations of location fingerprint components (Y is included; N is not included)

RSS	MED	RMSDS	AOA1	AOA2	AOA3	Mean error(m)
N	Y	Y	Y	Y	Y	<b>24.0416</b>
Y	N	N	Y	Y	Y	<b>22.0000</b>
Y	Y	Y	N	Y	Y	<b>28.9310</b>
Y	Y	Y	Y	N	N	<b>22.5832</b>

#### 4.3.3 Location target resolution

During the simulation, we assigned a grid of  $5\text{m} \times 5\text{m}$  pixels across the scenario and trained a location fingerprint for each pixel. Thus, the expected resolution can reach up to 5m. In this case, only one sample is being trained to identify one target. It is worth studying whether providing more samples for one target but sacrificing the resolution would enhance the training performance.

In Figure 4-16, mean errors are plotted against the resolution which is equivalent to the length of the side of each target square. For example, a  $10\text{m} \times 10\text{m}$  pixel contains four receiver grid points such that their centroid will be used as the target in which four sets of location fingerprints are trained towards. A 50m resolution suggests that 100 sets of training input are available for one location target. It can be seen that the performance degrades due to the spatial averaging and blurring of the fingerprint data.

From the trend of mean errors, we conclude that sacrificing the resolution to allow more training resources does not benefit the performance at all. Moreover, as the resolution increases the mean error also keeps increasing in general. It is not complicated to interpret that taking the centroid of a group of receivers is indeed wiping out individual information contained in each receiver location. Hence, the Neural Network is deemed to be capable of correctly handling the information being trained. It is not necessary for us to pre-process the input.

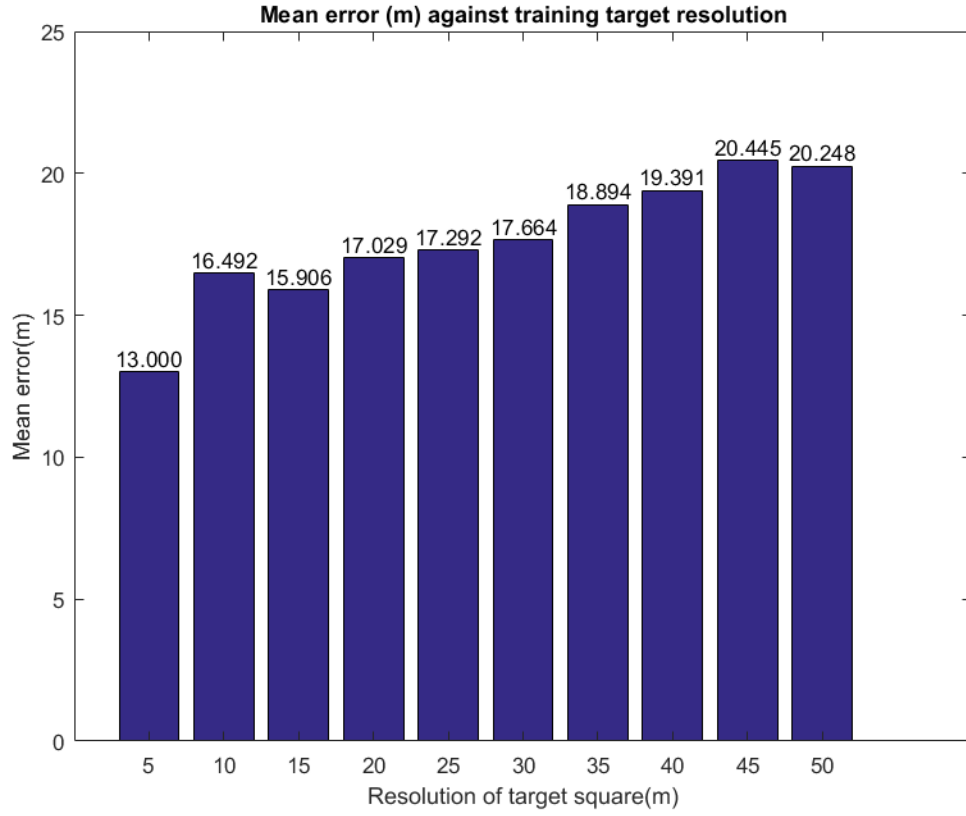


Figure 4-16: Mean error distance as a function of the pixel side length used in a training location.

#### 4.3.4 ANN configuration

The number of neurons for each layer influences the performance as well. A two-layer Neural Network has been chosen because it is believed to be more capable of handling nonlinear entries. Table 4.4 shows the network convergence performance, measured in metres of mean error, as a function of number of neurons in the 1st layer and 2nd layer. The range of varying neurons in each layer is set around half of its previous layer according to [26]. The result indicates the combination of 28 and 20 performs the best.



Table 4.4: Mean Error (m) against number of neurons in the 1st layer and 2nd layer

1st layer \ 2nd layer	24	26	28	30	32	34
10	20.668	19.966	19.740	20.390	19.854	18.476
12	19.645	20.545	18.785	19.286	18.950	18.407
14	20.265	18.766	20.027	18.857	20.081	18.720
16	20.122	18.941	17.907	19.741	20.840	19.450
18	20.414	19.412	20.458	18.663	20.646	18.383
20	18.658	19.684	<b>17.736</b>	18.156	18.396	18.806

Specifically, as the number of neurons in the 2nd layer increases the mean error mostly keeps declining and the 1st layer choice seems to make no difference. Generally, we realize that the more neurons allocated the better accuracy there is, though more neurons always takes an excessive amount of iterations and processing time. Through exhaustive searching, we determined that allocating 48 neurons in the 1st layer and 32 in the 2nd gives the smallest error, 13m. Since ANN training process is not supervised, it may produce networks of different mean errors given the same settings. Therefore, 13m of mean error is merely a nice figure which does not represent an optimised result.

Literally, the *pure linear* transfer function generates linear response which was found not to be suitable. The characteristics of common transfer functions are summarized in Table 4.2. Through trial and error, we determined that using *logistic sigmoid* for the first layer and *tangent sigmoid* for the second provided the best training performance.

Table 4.5: Training performance as a function of the assignment of training gain,  $Mu$ 

$Mu$	Number of iterations	Time(s)	Mean error(m)
0.01	6	99	13.0384
0.001	6	88	13.0000
0.0001	6	93	13.2665
0.00001	6	95	13.4536

We also tuned the training gain to reveal its effect on the performance. Table 4.5 shows how the training results are effected by the value of  $Mu$ . Since the number of iterations, processing time as well as the mean error are roughly the same, a conclusion is drawn that the training gain is not the saturating factor that ceases further training of the Neural Network. In other words, the Neural Network configuration is no longer able to provide better mapping performance.

#### 4.3.5 Noise level during measurement

In order to demonstrate the robustness of the mapping function, we deliberately added different levels of AWGN (Additive White Gaussian Noise) to the simulation results, as well as altered the environmental settings to generate corrupted fingerprints. Using these corrupted fingerprints, we attempted to prove that the user observations in the field do not have to be perfect to determine satisfactory localization.

AWGN, i.e., normally distributed random noise, is intentionally added to all  $15400 \times 6$  RSSs of the original location fingerprints in the Hoxton example. In the first trial, the mean of AWGN was set as zero to generate noise without bias. The standard Deviation ( $SD$ ) of the noise was tuned from 0.01 to 0.03 which equals 2.4 dB AWGN (the trained RSS values are truncated between -140 dBm and -60 dBm and normalised from -1 to 1) to reveal the impact of noise strength levels. As can be seen from Figure 4-17, the matching accuracies under  $SD = 0.01$  of noise are marked as a solid blue line in which approximately 90% of localizations are within 15 m of the true locations. At exactly 15 m of tolerance, the accuracy performance degrades from 75% to below 60% as noise strength increases from 0.8 dB to 2.4 dB on the solid yellow line and dashed magenta line.

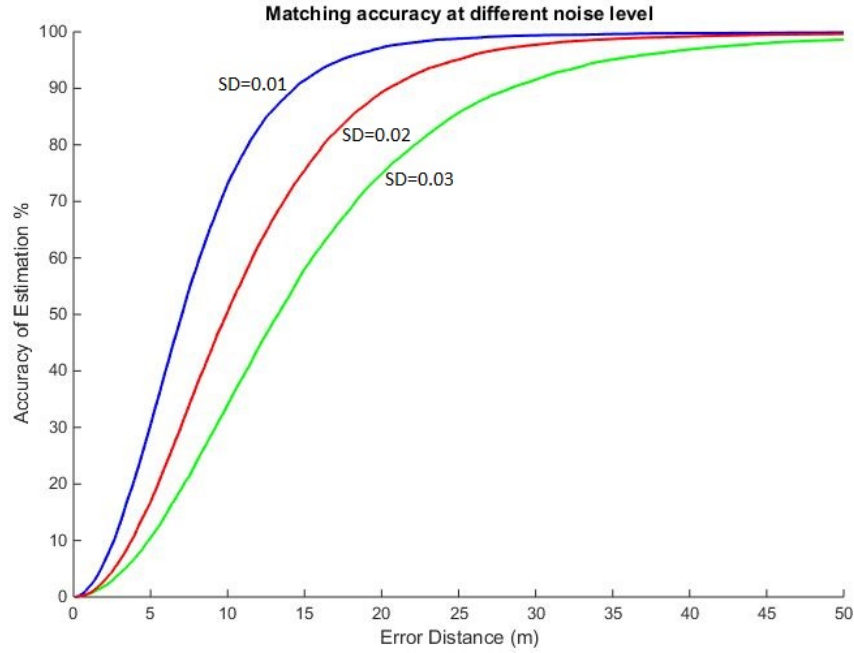


Figure 4-17: Cumulative Distribution Function (CDF) of localization accuracy against the standard deviations of the AWGN applied to the fingerprints.

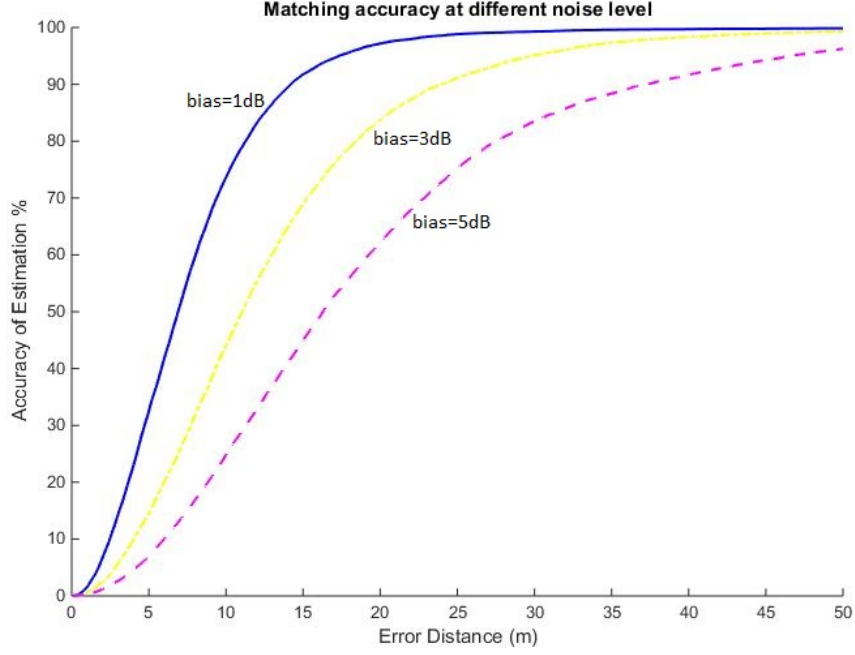


Figure 4-18: CDF of localization accuracy against measurement biases added to the fingerprints.

Matching accuracy under 1dB of biases are plotted in a solid blue line as shown in Figure 4-18. When different levels of biases up to 5dB are used to corrupt the fingerprints, the mean error performance is degraded from 15m to 20m, and up to nearly 30m at 90% acceptance rate. Generally, the mapping function proves to be immune to a degree of noise and biases added to the RSS observations.

#### 4.3.6 Environmental factors

Errors and deviations may be attributed to a limited representation of the environment. For example, the building shapes and surface properties are usually not available in full detail. The presence of traffic and pedestrians can also influence the channel. Furthermore, a reliable localization algorithm has to demonstrate universality in any urban environment.

In pursuit of answers for the above questions, we have applied the fingerprint extraction and training process to three different scenarios as listed in Table 4.6. In addition to the Hoxton example, the seven-object virtual scenario in Figure 3-12 and the Arc de Triomphe area in Figure 3-4 have been studied. Pixel size of  $1\text{m} \times 1\text{m}$  was assigned in the former. As in the Hoxton scenario, 5m squares were allocated in the Arc de Triomphe scenario since the scale is larger. Six opportunistic sources were used

in all cases such that the location fingerprint dimension is always 54.

Table 4.6: Training specifications and performances on different scenarios

Scenarios	Hoxton district	Seven objects		Arc de Triomphe
Scale(m)	$700 \times 550$	$120 \times 120$		$1000 \times 400$
Number of pixels	15400	14400		16281
Number of buildings	120	7		149
Number of sources	6	6		6
Mean error after training(m)	13.000	0.603		30.480
Type of disturbing variable		<i>A</i>	<i>B</i>	<i>C</i>
Disturbed mean error(m)		<b>1.791</b>	<b>2.686</b>	<b>38.508</b>

After training, 0.603m mean error is seen in the seven-object scenario; 17.136m is seen in the Arc de Triomphe scenario. We then added different types of error perturbations to the simulator to generate corrupted fingerprints, and fed them into the pre-trained Neural Networks (error free). The returned estimations were then compared with the authentic locations.

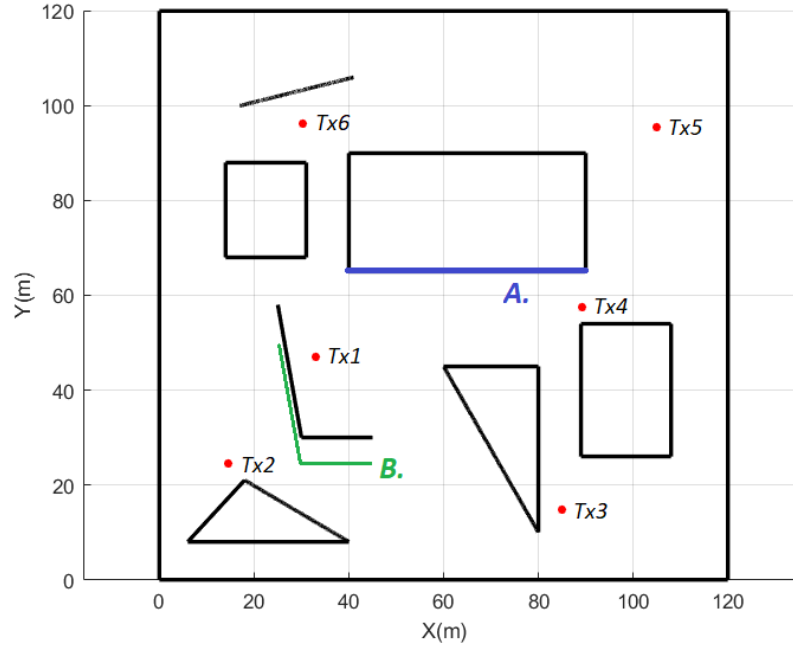


Figure 4-19: In test *A* the material of the wall highlighted in blue is manipulated; in test *B* the folded wall highlighted in green is moved downwards by 5 m.

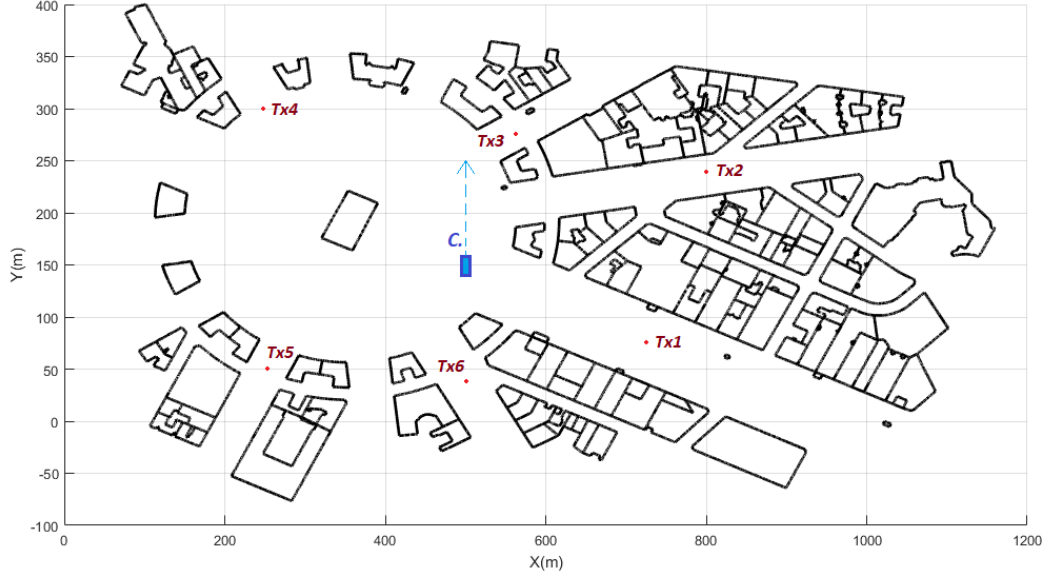


Figure 4-20: In test *C* a bus, of length 10m and width 3m, painted in solid blue is inserted into the Arc de Triomphe scenario and moved upwards for 10 consecutive steps.

In test *A* shown in Figure 4-19, we changed one of the longest surfaces from concrete, with relative dielectric constant 4.0 and conductivity 0.0001 S/m, to glass whose dielectric constant equals 7.0 and conductivity equals 0.005 S/m. As a result, the estimation error rises from 0.603m to 1.791m. In test *B*, a folded wall was moved downwards by 5m. This time the estimation error increases from 0.603m to 2.686m.

In test *C* on the Arc de Triomphe scenario, we put a double decker bus at the junction shown in Figure 4-20. The center of the bus was at (500m, 150m). The shape of the bus was projected as a rectangular, of 10m×3m, at which its height is equivalent to average roof height in the scenario. Consequently, the MSE increases from 30.480m to 38.508m.

In order to demonstrate the trained network is generally reliable to moving obstacles at arbitrary location, we made the bus move on an upwards trajectory for another 10 steps as shown by the arrow in Figure 4-20. The MSE results relying on the originally trained network are listed in Table 4.7. The mean error hardly deteriorates for more than 10m. Notice that as the bus moves away from the main street (Champs-Élysées Avenue) the mean error actually drops. Presumably, we believe reflections occurring at a crucial location, e.g., the junction of a thoroughfare, would make a more significant difference to the propagation paths which definitely influence the authenticity of fingerprints significantly.

Table 4.7: Mapping performance using the trained network in the presence of a moving double decker bus

Bus position on $y$ axis(m)	Mean error(m)	Degradation(m)	Percentage
150	38.508	8.028	26.34%
160	41.201	10.721	35.17%
170	38.515	8.035	26.36%
180	40.021	9.541	31.30%
190	40.884	10.404	34.13%
200	35.986	5.506	18.06%
210	34.937	4.457	14.62%
220	34.989	4.509	14.79%
230	34.844	4.364	14.32%
240	34.823	4.343	14.25%
250	34.834	4.354	14.28%

Nevertheless, the mean error increments among these tests are never significant. It is generally concluded that the localization algorithm does show a considerable degree of robustness against various environmental uncertainties.

## 4.4 Summary

In this chapter, a location fingerprint is firstly defined based upon channel modelling results. These were then applied to ANN to train the location fingerprints to their corresponding 2D coordinates. This has been shown to generate a deterministic network leading any user observation towards an estimated location. The accuracy and reliability of this method have been validated against multiple errors and perturbations for different scenarios.

## Chapter 5

# Source localization

The localization of sources has drawn increased attentions recently, especially in high-rise environments where the traditional triangulation methods become unreliable due to multipath propagation. The prospective applications of such localization techniques range from commercial navigation, criminal investigation, to military defence systems.

In Chapter 4 we discussed and validated a novel fingerprint-based localization approach. A ray launching model has been developed and shown to be capable of providing channel estimations with respect to the observing location. In this chapter, we will now investigate whether fingerprints of unknown sources can be defined to aid a source localization solution. An appropriate matching algorithm is required in order to find the best match within a pre-computed fingerprint database.

This chapter details the idea of a source location fingerprint as well as the development of a matching algorithm. Case studies based on real-world environments present validation of this method, and simulations exposed to a variety of perturbations reinforce the reliability of this method. Of particular interest in this chapter is the use of UAVs (Unmanned Aerial Vehicle) for the collection of raw data to be matched. We also put forward several concepts, particularly the UAV route strategy, to improve the reliability to meet the needs of practical applications.

### 5.1 Source localization algorithm

Location of unknown radio sources in urban environments is of great research interest currently. The traditional localization approaches are found to be less effective in severe multipath environments [31, 41]. Many researchers in the literature have proposed to determine geographical location of radio sources by monitoring several

spatially dispersed sensors [79, 92]. Lohrasbi-peydeh *et al.* [78] proposed a location decision algorithm using RSS as indicator. Adelipour *et al.* [5] proposed a source localization method using TDOA and FDOA (Frequency Difference of Arrival). Shahidian *et al.* [112] utilized a mobile aerial sensing device to collect spatially dispersed data. Using a dense network of sensors the localization is essentially transformed into a compressed sensing problem from which a deterministic solution can be found. However, instalment of the sensing devices becomes a pre-requisite, and the quality of the data becomes critical which is usually non-ideal in busy city streets [128].

To mitigate the above issues, here we propose to make use of a mobile payload carried by a UAV, to conduct continuous measurements along ‘street canyons’. From these measurements, characteristic fingerprints can be extracted to discriminate potential source locations. The key steps to accomplish this source localization are the development of a practical measuring scheme and a robust matching algorithm.

### 5.1.1 Data collector

In order to capture the spatial electromagnetic features that discriminate source locations, the joint contributions of as many pieces of data as possible from the field measurement is preferred rather than relying on persistent observations in fixed locations. An example of such data collection strategy is illustrated in B of Figure 5-1. This is also a more realistic assumption for the ray launching model accuracy since trends of variation are easier to predict than an absolute channel response.

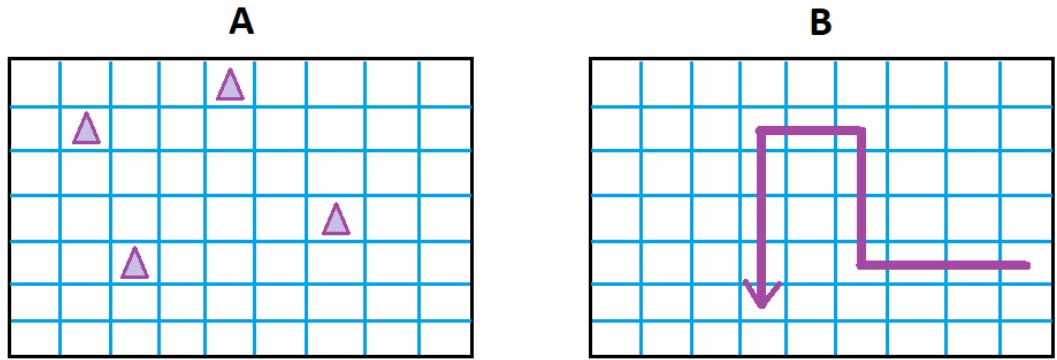


Figure 5-1: A shows a conventional compressed sensing method in which sensors, marked as purple triangles, are dispersed across the scenario at fixed locations; B demonstrates a data collection strategy along a specified path which contains more spatial dispersion information of the received parameters.

Following the idea of remote sensing [122], we note it is indispensable to have a



mobile measurement device which is able to go along a specified path with constant speed and wide coverage. A small drone is, therefore, an efficient tool to collect much information in a short period [37, 126]. By careful selection of the flight path which may be constrained due to building height or for security reasons, a data series comprised of electromagnetic observations can be recorded to leverage the fingerprint extraction.

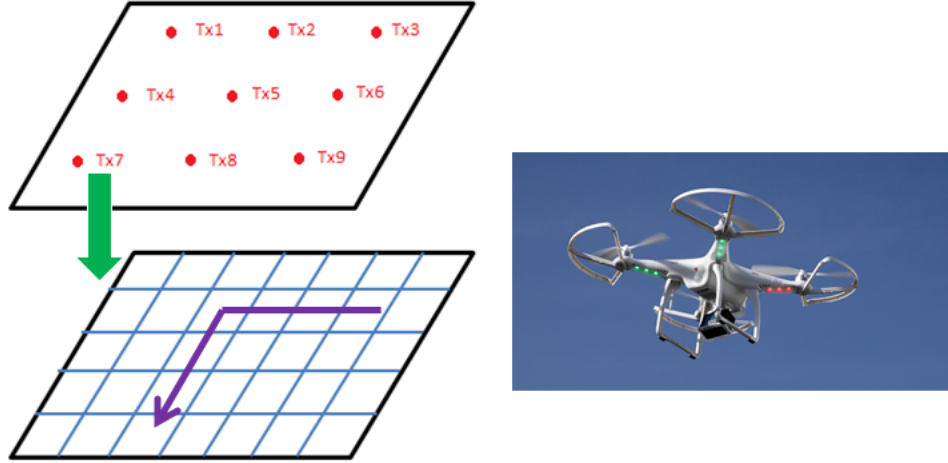


Figure 5-2: An illustration on how to use a UAV to collect data from the field: the UAV moves along a route marked in purple in the urban environment; its observations can be simulated using the ray launching model given transmitter source candidates (Tx1, Tx2, Tx3, ... Tx9).

The fingerprint obtained above would be an input enquiry to the algorithm which seeks to determine the source location during the “n-line” query phase. The “off-line” job is one of exhaustively simulating what the drone would see along that path with respect to a grid of source locations. From these simulations, a look-up table can be constructed forming the fingerprint database for all candidate locations.

As can be seen from Figure 5-2, a quad-rotor drone can be used to carry out a measurement campaign along a route designated by the purple arrow. The blue network of grid locations implies that the electromagnetic characteristics at each location can be simulated from the ray launching model. The red dots in the upper layer denote the emitter candidates which are very likely to generate disparate observations along the route. In principle, it is the simulated observations along that route which lead to fingerprint extraction, so as to give rise to source localization.

Hence, the UAV is playing the role of data collector by conducting continuous sampling ideally at a constant speed and at a fixed height. A desirable route is expected to wind through street canyons and provide a wide-spread coverage in which the UAV

can ideally thread across LOS and NLOS areas from different perspectives to create variations in the received parameters. Further discussion on route selection is given in section 5.2.3.

### 5.1.2 Feature transformation

Given a time series of simulated receiver observations along a specified UAV route, a link between the received characteristics and transmitter source locations is made. Yet this alone is not adequate to make a source localization algorithm unless the features can be explicitly and numerically recognized. A feature transformation process is necessary to quantize both the UAV observation and the simulated observations [53].

As has been discussed in Chapter 4, a location fingerprint may contain a combination of normalized RSS, TOA and AOA which are derived from Eqn. (4.1), Eqn. (4.2), Eqn. (4.3), Eqn. (4.4). A typical field measurement of RSSs is shown in Figure 3-16. Superficially, variations over short distances are likely to be attributed to noise and fast fading which are less valuable, whereas, variations over longer distances tend to reveal genuine information on path loss.

A significant complicating factor is that the UAV may not be perfectly synchronized in time and position. That is, the route may deviate from the expected one. For example, suppose the UAV measures some parameter with a sinusoidal variation such that time variation is shown in *A* of Figure 5-3. Consider now, a time delay during measurement results in a phase shifted red curve in *B*. Inaccurate attenuation or surface reflectivity assignment may make amplitude distortion apparent as shown in *C*, while noise and fast fading would corrupt the curve as shown in *D*.

To counter these problems, the basic idea of feature transformation is to project the observed data series onto a lossless domain in which periodic characteristics are emphasized, whereas, the phase and amplitude can be stripped.

A Discrete Fourier Transform (DFT) happens to be such a lossless transform and is able to decompose a data series into a number of periodic components. The  $k$ -th DFT coefficients,  $X_k$ , can be written as;

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-j2\pi kn/N} \quad (5.1)$$

where  $N$  is the number of samples and  $x_n$  represents the  $n$ -th sample value. Both real and imaginary parts of the frequency components are taken as feature eigenvector.

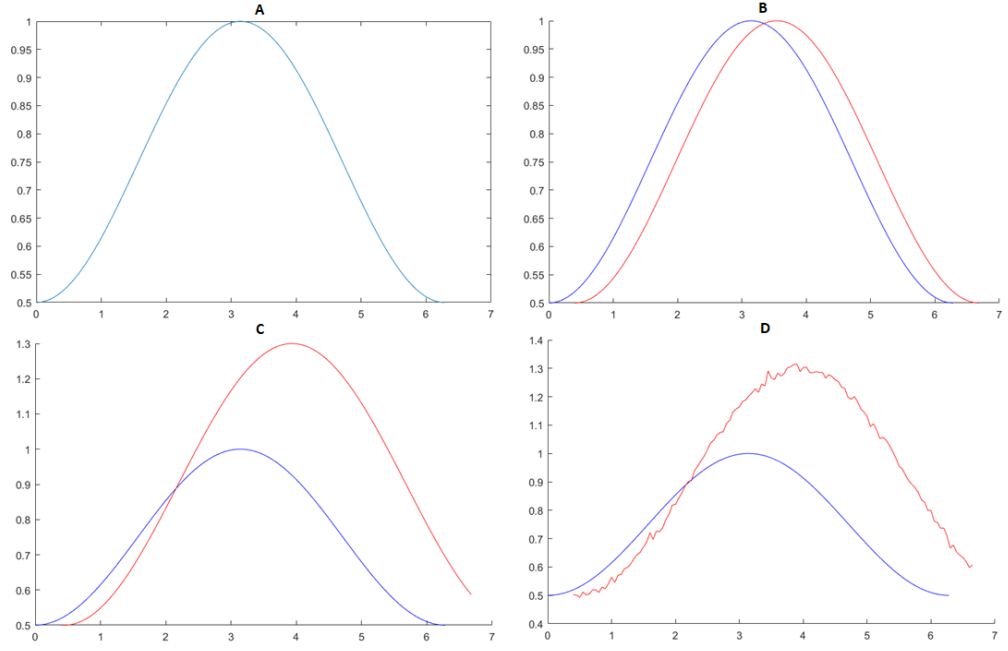


Figure 5-3: An illustration of potential data time-series corruptions: case A shows the original waveform in blue; case B shows the waveform after phase shift which is marked in red; case C shows the waveform after amplitude distortion which is marked in red; case D shows the waveform exposed to AWGN which is marked in red. (Scales are in arbitrary units)

After applying the DFT to reveal spatial frequency features, a Low-Pass Filter (LPF) can be used to reduce or eliminate high frequency components which are most possibly caused by noise and fast fading. A simple example of such LPF may have a gain equal to unity below the cutting-off frequency and zero above. The filtered spectral components form the feature vectors for the specified route.

Through these steps, the features of the source locations are transformed into lower frequency coefficients in frequency domain from the original time-series. These feature vectors are stored in a look-up table to be checked against an input vector which is to be obtained from field measurement through an identical measurement procedure.

### 5.1.3 Source location fingerprint

In order to identify the source location accurately, the ‘fingerprint’ database, which contains feature eigenvectors extracted from the simulated UAV observation, should be built for a sufficiently large number of source candidate locations. These source candidates are also allocated in a grid-like fashion with a uniform separation distance equivalent to the localization resolution [20]. For each simulated source location, the

discrete observations along the UAV route should be concatenated into a data series. We aim to demonstrate that these electromagnetic observations are not only measurable but possess physical meanings that can be used to discriminate transmitter locations.

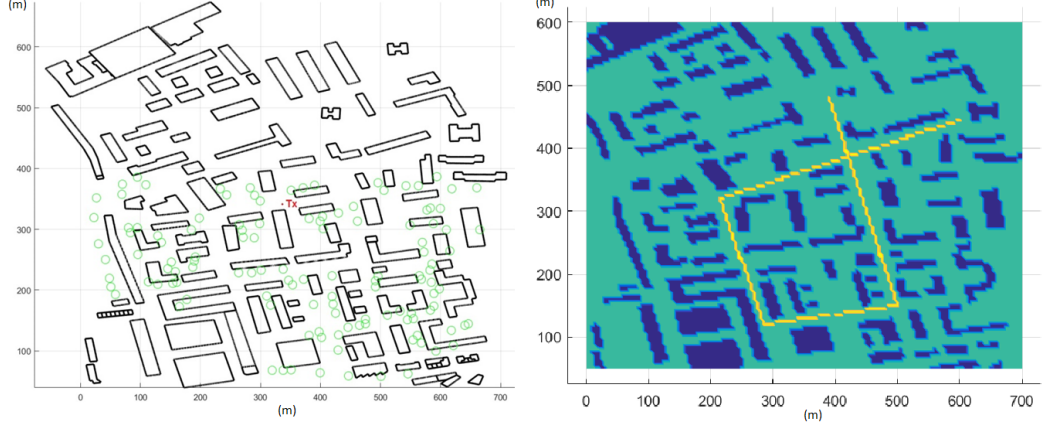


Figure 5-4: Simulated UAV route for the Hoxton scenario where the route is highlighted in yellow and buildings are depicted in dark blue.

With reference to the Hoxton district, an example UAV path is highlighted in yellow as shown in Figure 5-4 (right). The high-rise buildings are depicted in dark blue over which it is forbidden to fly the UAV. The green pixels show many obvious street canyons. Obtaining all receiver characteristics allows more flexible route planning as well as freedom in analysing an optimal route in terms of the feature transformation process.

By performing the simulation and extraction process for all source locations, a source location fingerprint can be determined eventually consisting of the lower spatial-frequency components which contain the varying trends of RSS, MED, RMSDS and AOAs. An input measured fingerprint is obtained following the same approach and used to search against the simulated fingerprint database to estimate the source location. The final step to success is a matching algorithm to find the best match between the measured fingerprint and those simulated fingerprints in the database.

#### 5.1.4 Matching algorithm

A commonly used algorithm that could be applied to measure eigenvector similarity is the Euclidean distance. It considers two  $n$ -component vectors as two points in  $n$ -dimensional space and calculates the root mean square of their distance in that abstract space [58]. In a Cartesian coordinate system, an expression of the Euclidean distance

between  $\vec{a} = (a_1, a_2, \dots, a_n)$  and  $\vec{b} = (b_1, b_2, \dots, b_n)$  can be written as;

$$\|\vec{a} - \vec{b}\| = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \quad (5.2)$$

where  $n$  denotes the dimension. Other common distance measures include discrete metric, Manhattan distance metric (also referred to as Taxicab distance), graph metric etc. These metrics have been examined by using a measurement-based fingerprint to find best match among the database of simulated ones.

However, disappointingly, the test results show that a simple Euclidean distance metric, or the above metrics, do not yield an accurate matching algorithm. Both measurement errors and flaws in these approaches may account for such failures. Measurement errors include the interferences of noise and fast fading, measurement route deviation due to wind effects, poor synchronization as well as the fact that the sampling may not be evenly distributed along the route as expected.

A critical mechanism flaw with the above metrics concerns the significance of each dimension. Whereas, in reality the unexpected and unknown phase and amplitude distortions permit the bins to shift along the frequency axis, which makes dimensional alignment problematic. To mitigate this problem, the ideal matching algorithm must possess a flexible pattern recognition scheme which is tolerant of distortions and misalignments.

According to [56, 72], the Dynamic Time Warping (DTW) approach is ideally suited to this kind of problem. Given two data series  $\vec{a} = (a_1, a_2, \dots, a_n)$  and  $\vec{b} = (b_1, b_2, \dots, b_m)$  of length  $n$  and  $m$ , respectively, an  $n$ -by- $m$  matrix can be constructed in which the  $(i, j)$  element contains the distance between two points  $a_i$  and  $b_j$ . Each matrix element corresponds to the alignment between points  $a_i$  and  $b_j$ . A warping path  $W$  is a contiguous set of matrix elements that define a mapping between  $\vec{a}$  and  $\vec{b}$ . The  $k$ -th element of  $W$  is defined as  $w_k = (i, j)_k$ . Hence, the DTW is the path that minimizes the warping cost given by

$$\text{DTW}(\vec{a}, \vec{b}) = \min \left\{ \sum_{k=1} w_k \right\}$$

If the fingerprints to be compared are regarded as data series  $\vec{a}$  and  $\vec{b}$ , the path that leads to the minimum warping cost corresponds to the best alignment. Then Euclidean distance, or other distance metrics, can be applied to calculate corresponding distance. It should also be noted that DTW can deal with inputs of different lengths. Further note that an important variable of DTW, named as width of adjust-

ment window, restricts the warping path to be within a specific number of samples of a straight-line fit. The effect of distance metric used and width of adjustment window in terms of DTW are discussed in sections 5.2.1 and 5.2.2, respectively.

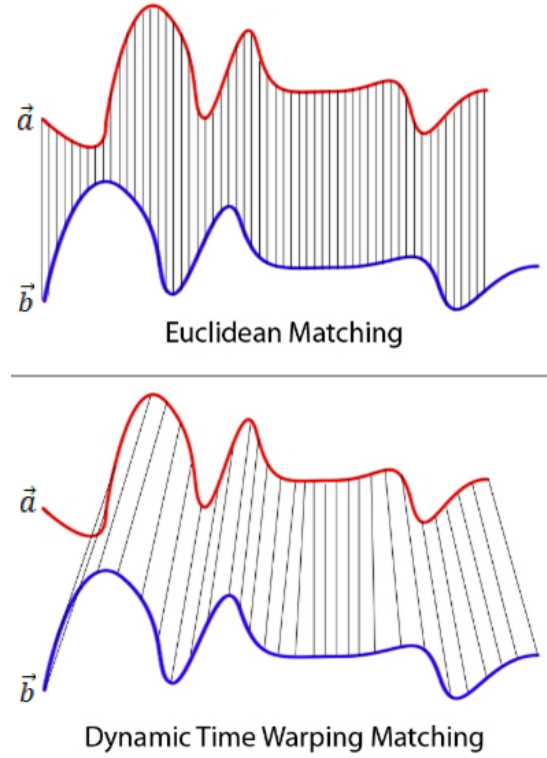


Figure 5-5: A comparison between Dynamic Time Warping and Euclidean matching.

Specifically in our fingerprint recognition problem, the DTW algorithm is able to exhaustively seek for the best match from a discrete data among its neighbours, so as to determine an optimal alignment between the two data series. An illustration on how DTW works superior than Euclidean matching is shown in Figure 5-5. Using Euclidean distance, the blue envelope is going to find a large difference to the red. On the other hand, DTW overcomes the stretching distortion and acknowledges their similarity.

In this final step, the DTW is assisting us in retrieving similarity measures between the sought fingerprint and simulated source location fingerprints. The best match will be returned as the estimated source location. Further accuracy and reliability analysis are detailed in the next section.

## 5.2 Accuracy of analysis

A data mining approach for fingerprint extraction and localization has been proposed. Nevertheless, the practical situations are rarely as straightforward. Several factors may undermine the effectiveness of the approach, e.g., limited accuracy of the ray launching model even though it has been validated, uneven sampling positions along the route, wind conditions during measurement, parameters of the matching algorithm, as well as the route selection strategy which generates the fingerprint. These factors can be generally divided into two categories: the noise and deviations during measurement and simulation; the UAV route selection which triggers location determination.

Table 5.1: Case studies for two real-world scenarios

Case study	Hoxton district	Munich downtown
Scale	750m $\times$ 550m	2400m $\times$ 3400m
Number of receiver pixels	15400	54471
Pixel size	5m $\times$ 5m	10m $\times$ 10m
Number of objects	120	2088
Number of walls	1376	34890
Number of trees	123	1758
Number of candidate emitters	130	616
Number of samples	234	970;355;1031
Expected resolution	50m	100m
To be studied	Noise & perturbations	Route strategy

In order to verify the source localization theory, we carried out tests on two real-world scenarios. As is shown in Table 5.1, a typical residential area in Hoxton district, London, and the Munich city center area were investigated. Both of them come with serried buildings and trees while having broad coverage with precise simulated receiver pixels. In this study,  $13 \times 10 = 130$ , and  $22 \times 28 = 616$  emitter candidates, respectively, were assigned in grids across the scenarios to give resolutions of 50m and 100m. In the Hoxton district, the aim is to mainly concentrate on the accuracy performance in the presence of various perturbations. In the Munich scenario, the aim is to primarily study which one among the three measurement routes is the best for fingerprinting and how to select an optimal UAV route in any urban environment.

### 5.2.1 Measurement error and path deviation

A UAV route winding through street canyons is marked in yellow as shown in Figure 5-6 (a). The UAV is assumed to have a constant velocity of 1 m/s and a sampling rate of 0.2 samples per second. Thus, it is able to take a sample every five metres which guarantees a continuous sampling trace comparable to the receiver resolution. Since there are 130 candidate emitter locations of which each has a signature fingerprint, we would like to test how much degree of interference a fingerprint could bear to still successfully determine the true emitter location.

The scenario where the UAV only uses half the sampling rate (0.1 samples per second) which gives rise to skipping samples is shown in Figure 5-6 (b). In Figure 5-7 (c), the first turning way point is slightly moved towards the right to imitate the wind effect blowing to the right during first half of the journey. In Figure 5-7 (d), the UAV just keeps making zig-zag sampling decisions along the path to imitate the presence of random directional gusts which unexpectedly deviate the measurement route.

Equivalent perturbations were applied to the originally simulated electromagnetic primitives in order to create corrupted observations which give rise to noisy and deviated fingerprints. The accuracy performance is a measure of how many among the 130 corrupted fingerprints can still be used to determine the authentic source locations. Table 5.2 lists the accuracies against different kinds of perturbations in which the observed RSS, TOA and AOAs are all subject to 4dB AWGN.

Table 5.2: Accuracies of relying on RSS, AOA1X, AOA1Y, MED or RMSDS subject to various simulated measurement errors

Accuracy(%)		RSS	AOA1X	AOA1Y	MED	RMSDS
AWGN=4dB	Directional wind	95.38	62.31	67.69	66.15	63.85
	Random wind	92.31	50.00	48.46	53.08	54.62
	‘Manhattan’ metric	100	100	100	96.15	94.62
	Half sampling rate	100	100	100	94.62	90.00
	Half LPF cut-off	100	100	100	96.15	88.46



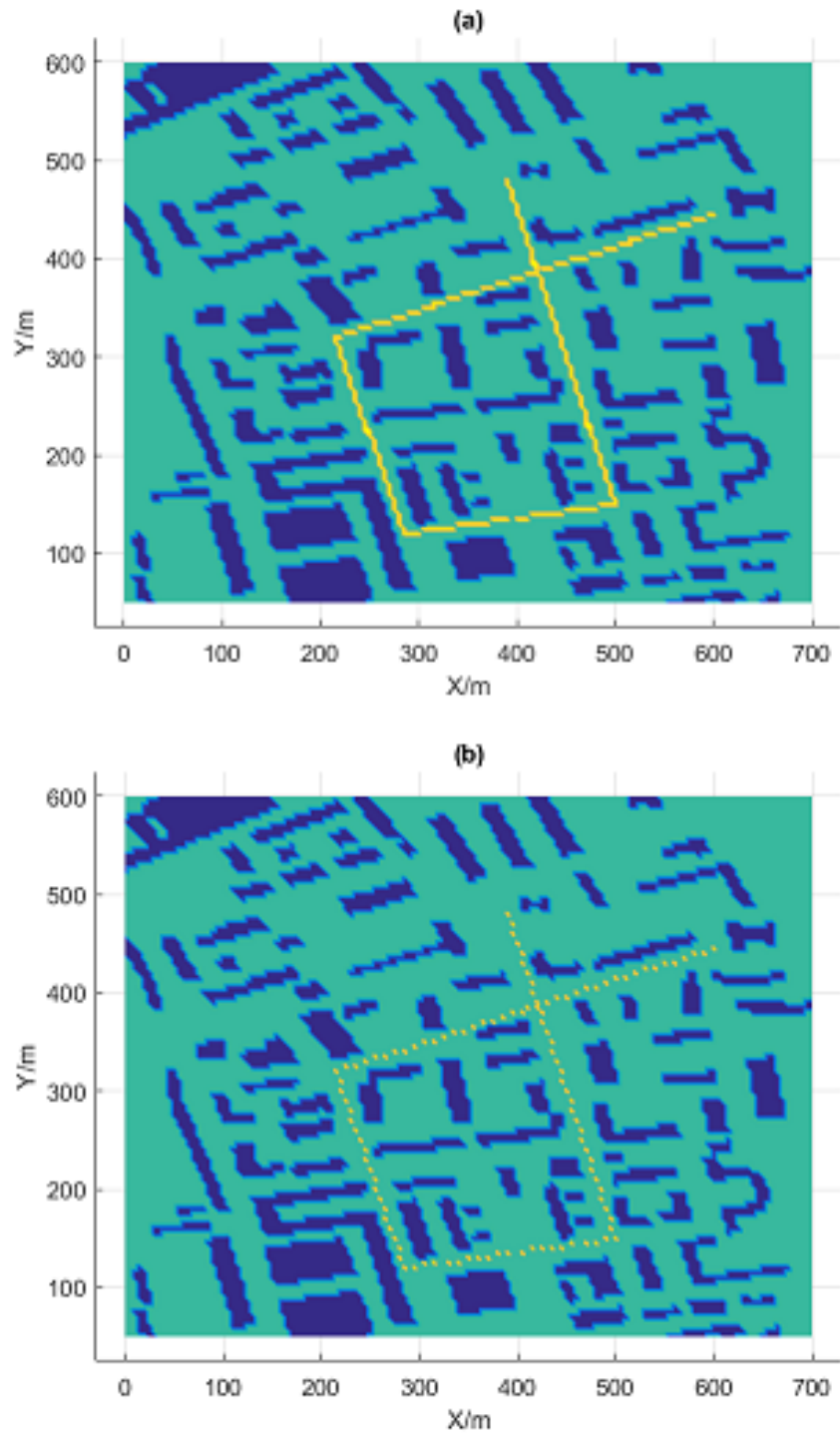


Figure 5-6: Simulated UAV route deviations applied to the Hoxton scenario: *a)* Original route; *b)* Halved the original sampling rate.

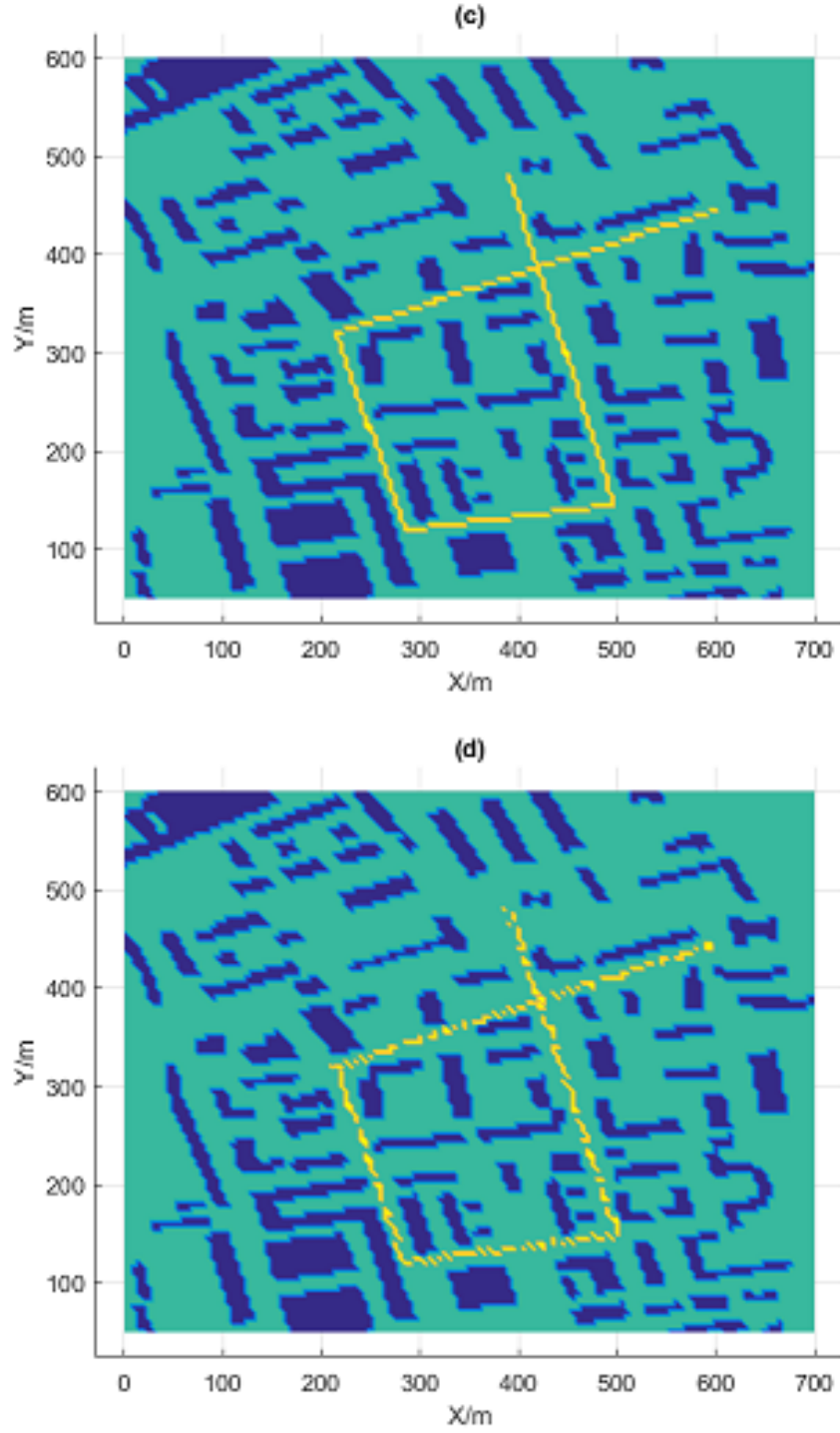


Figure 5-7: Simulated UAV route deviations applied to the Hoxton scenario: *c)* Directional wind blowing where the first turning way point is shifted; *d)* Random-directional wind blowing when the UAV is making zig-zag sampling decisions along the route.

Note that the accuracies are obtained in Table 5.2 entirely relying on one observed electromagnetic primitive at a time. It can be seen that the RSS appears robust to all kinds of corruption compared to the other primitives. The RSS remains over 90% accuracy in either kind of windy condition, and returns perfect estimations for all types of error. Except for wind effect, the trends of variations of AOA1X or AOA1Y are also found to be reliable. Whereas, time delays are not very good fingerprint primitives to discriminate source locations.

In fact, the behaviour of the accuracy is to be expected because the fingerprint extraction method has been, in a sense, specifically tailored for RSS. Imagine a boundary of LOS and NLOS areas separated by the shadow of a wall. The RSS variations of variation tend to be smooth and continuous due to diffractions on the vertical edges of the wall. The feature transformation utilizes DFT and a LPF to extract large spatial variations as a fingerprint eigenvector which favours continuously changing primitives such as RSS. The TOA and AOA variations possess more discontinuities in comparison. On the other hand, it is usually the easiest to measure RSS with simple mobile equipment fitted on a UAV.

It can clearly be seen that windy conditions have the most dramatic impact on accuracy performance compared to other variables. Under randomly blowing wind, only the RSS stays reliable. All primitives other than RSS are severely affected by directional wind, they still provide good information on the true source locations. Using a hybrid method, e.g., KNN, combining multiple parameters including AOA1X, AOA1Y, MED and RMSDS may be a promising approach to better determine the authentic source location.

In section 5.1.4, we have introduced a DTW algorithm in which a Euclidean distance metric is used to calculate similarity between two realigned data series. The ‘Manhattan’ metric has also been studied. The results show that the localization accuracy remains outstanding. Additional distance metrics have been studied including the modification of the power of  $(a_i - b_i)$  in Equation 5.2 from 1 to 3 in 0.1 increment. These modifications have little effect on the results. Note that the fingerprint elements defined can be non-physical such that the matching procedure does not have to make physical sense but can only be judged good or bad by its resulting accuracy.

Halving sampling rate is, in essence, getting rid of every other discrete measurement sample. In the frequency domain, exactly half of the bins are available this time and the higher frequency components are omitted. Therefore, it is actually a question of to what distance does a spatial variation make a difference.

Applying a LPF with one half of the cut-off frequency gives a similar result. An ideal LPF will only keep the DFT coefficients below the cut-off frequency. Given the

sampling frequency here  $F_s = 0.2\text{Hz}$  and the total number of samples along the route  $N = 234$ , each frequency bin can be written as

$$f_{bin} = \frac{F_s}{N} = 8.55 \times 10^{-4}\text{Hz}$$

If  $n$  is the cut-off frequency which equals the number of bins included and the UAV speed is  $v = 1\text{m/s}$ , the smallest detectable distance,  $D$ , of variation becomes

$$D_{\min} = \frac{v}{n \cdot f_{bin}} \approx \frac{1170}{n}$$

Originally only spatial wavelengths of over 50 m are taken as a fingerprint signature, so solving for  $n$  gives

$$\begin{aligned} \frac{1170}{n} &\geq 50 \\ n &\leq 23.4 \end{aligned}$$

Thus,  $23 \cdot f_{bin}$  is set as the LPF cut-off frequency. If it is halved, the 11 lowest frequency components are taken as a fingerprint. As can be seen in Table 5.2, the results show that halving the LPF cut-off frequency does not significantly affect the performance.

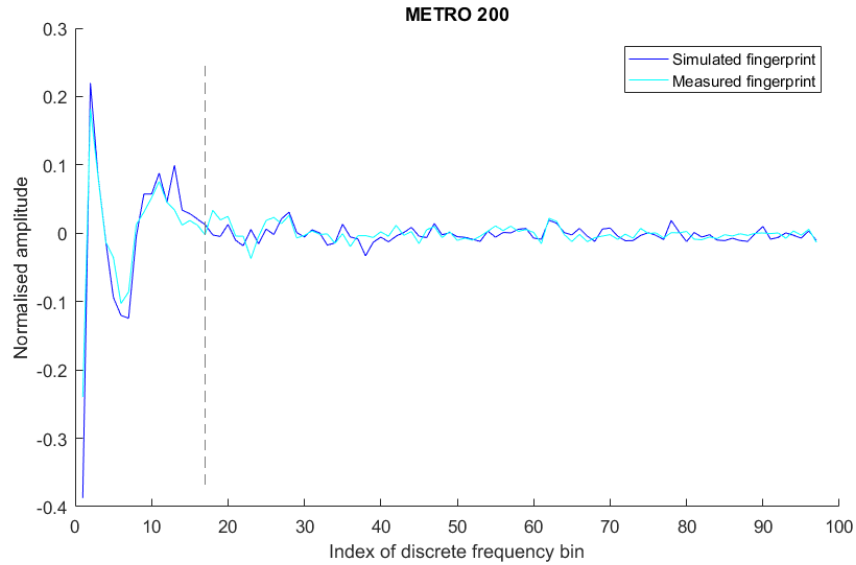


Figure 5-8: Light blue envelope represents measured fingerprint of route ‘METRO 200’; blue one is the corresponding best match found among the fingerprint database. As can be seen significant amplitudes are all located at the lower frequency end.

The effect of varying the cut-off frequency has been investigated by considering as few as two to up to 100 frequency bins. By way of example, Figure 5-8 shows simulated

and measured spatial frequency amplitudes along a UAV route (further details of the route and measurement are given in section 5.2.3). It can clearly be seen that only the lowest frequency components have significant amplitude. Typically it has been found that as long as more than five bins are included the accuracy is always very good. This implies that spatial variations over  $\frac{1170}{5} = 234\text{m}$  make the greatest contribution to source location discrimination. Furthermore, the absolute amplitude values of higher frequency components are in fact much smaller than lower ones. Even though small spatial variations down to  $\frac{1170}{100} = 11.7\text{m}$  wavelength are taken into a fingerprint, the high frequency fingerprint elements make little difference to the matching result.

### 5.2.2 Noise level and bias

Deviations may not only come from measurement errors but also the inevitable background noise and environmental perturbations. Path loss factors, reflectivity of building surfaces, fast fading, unexpected obstacles etc. cannot always be accurately represented. The localization algorithm must possess a degree of robustness against noise and biases contained in the observations.

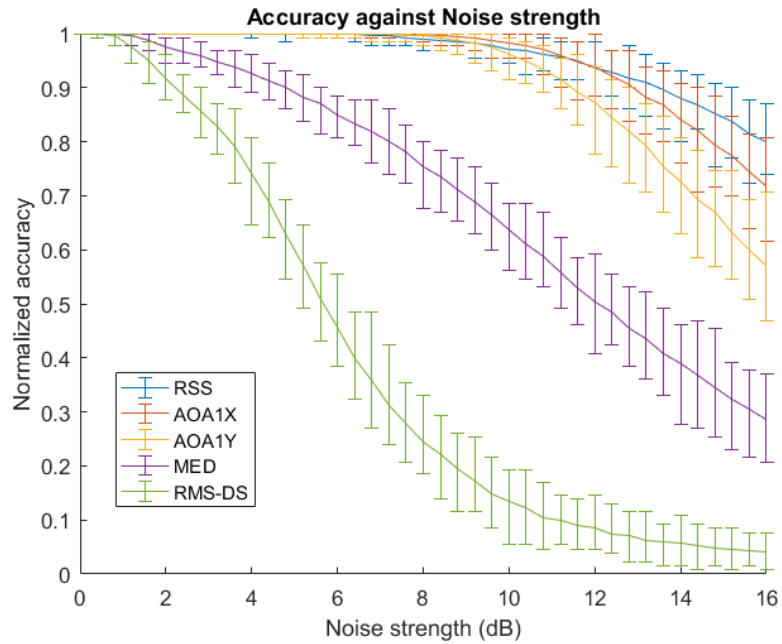


Figure 5-9: Accuracies of relying on RSS, AOA1X, AOA1Y, MED or RMSDS against noise strengths; the curves represent the average of 100 simulations, and the error bar represents the minimum and maximum.

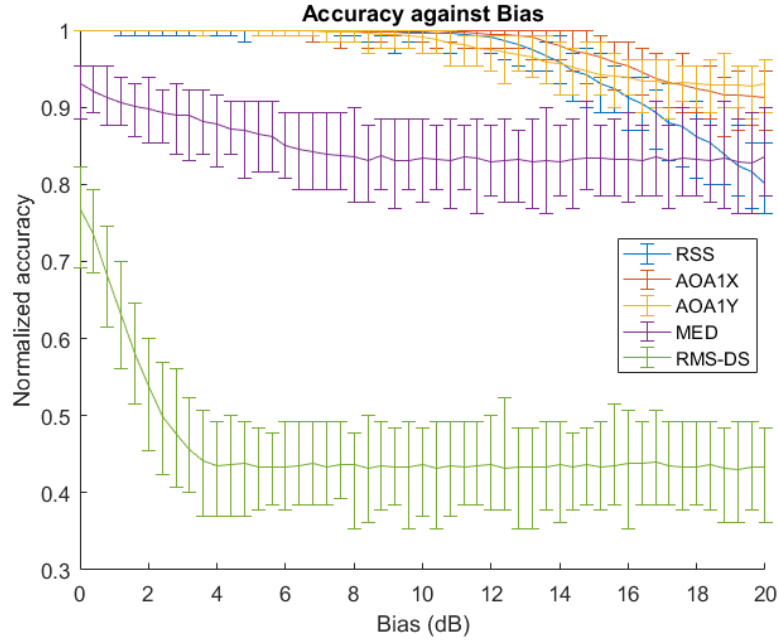


Figure 5-10: Normalized accuracies of relying on RSS, AOA1X, AOA1Y, MED or RMSDS against biases; the curves represent the average of 100 simulations, and the error bar represents the minimum and maximum.

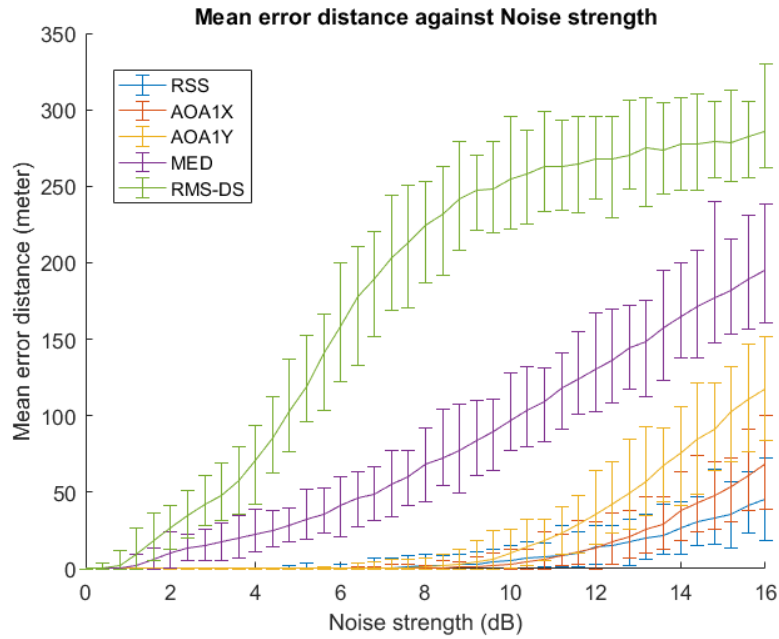


Figure 5-11: Mean error distances of relying on RSS, AOA1X, AOA1Y, MED or RMSDS against noise strengths; the curves represent the average of 100 simulations, and the error bar represents the minimum and maximum.

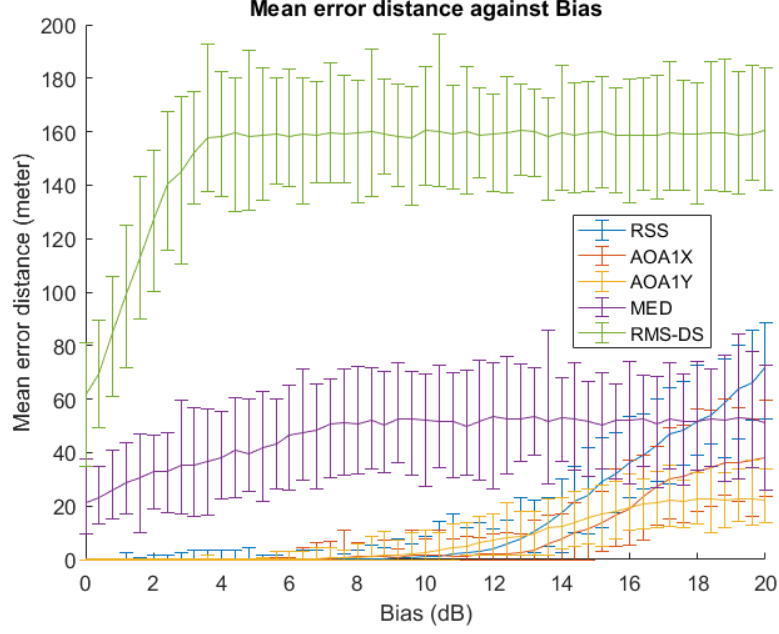


Figure 5-12: Mean error distances of relying on RSS, AOA1X, AOA1Y, MED or RMSDS against biases; the curves represent the average of 100 simulations, and the error bar represents the minimum and maximum.

Therefore, we tested the accuracy performance with respect to increasing levels of noise strength and bias applied to the simulated measurements as shown in Figures 5-9 and 5-10. To generate representative statistics, 100 sets of normally distributed noise are generated for each level of strength or bias. The curves show the average values and the errorbars represent the greatest (minimum and maximum) deviations found. The accuracies were determined solely relying on only one electromagnetic primitive at a time. It can be seen that the RSS, AOA1X, AOA1Y all demonstrate impressive robustness in the presence of noise. Under 10 dB AWGN, all of them produce results with over 95% accuracy. The MED is less reliable, whereas, the RMSDS is not trustworthy at all.

On the other hand, RSS, AOA1X and AOA1Y also perform extraordinarily well against biases. All of them return over 90% accuracy at 16 dB bias which is enormous. This performance, in the presence of such a large bias, validates the effectiveness of DTW which is able to recognize similarities between data series regardless of the amplitude gap. The performance of MED is acceptable, whereas, RMSDS is totally ineffective at the presence of bias.

Figures 5-11 and 5-12 reflect the performances in terms of mean error distance. Below 10 dB of strength, the AWGN is not able to deteriorate the mean error by more

than 20m if RSS, AOA1X or AOA1Y is being used. By contrast, RSS, AOA1X and AOA1Y can tolerate a bias up to 14dB and achieve the same performance.

It can be seen that AOA1X outperformed AOA1Y both in terms of localization accuracy and mean error distance. Since their values are Cartesian projections of one vector, one might expect them to have the same performance. However, this is not the case since the results are route dependent. This has been investigated by reversing the coordinate system in the Hoxton scenario in which the length, 700m, used to be larger than the width, 550m. Based on a symmetrical UAV route and identical transformation procedures, the analysis was repeated. The accuracy and mean error distance against the increasing level of noise strength for 100 different sets are shown in Figures 5-13 and 5-14. It can be seen that AOA1Y outperformed AOA1X which implies that superiority of AOA1X compared to AOA1Y depends on the geometry. Since the area of interest is a rectangular, a parameter which has more significant variation along the longer side possesses better spatial dependency, allowing improved fingerprint discrimination. This explains why the Cartesian projection of AOA along the longer side of the scenario always produces slightly better performance.

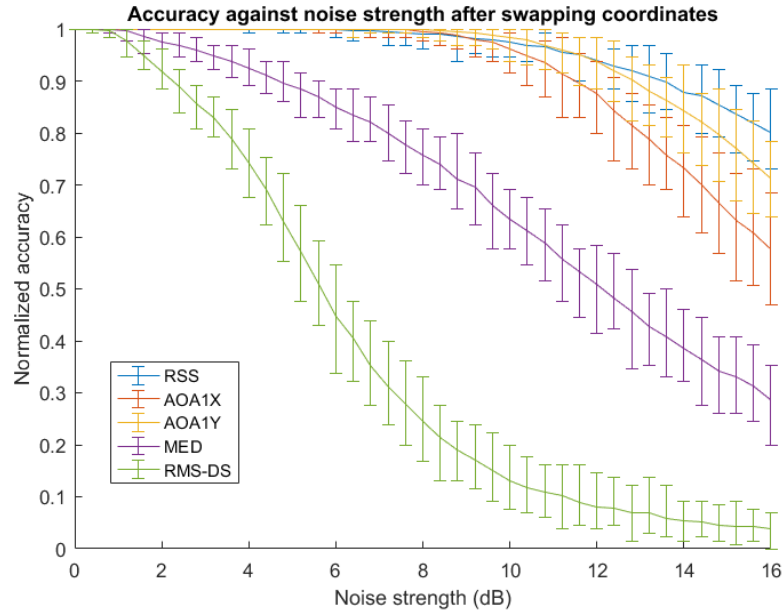


Figure 5-13: Accuracy of relying on RSS, AOA1X, AOA1Y, MED or RMSDS against biases after reversing the coordinate system of the Hoxton scenario; the curves represent the average of 100 simulations, and the error bar represents the minimum and maximum.



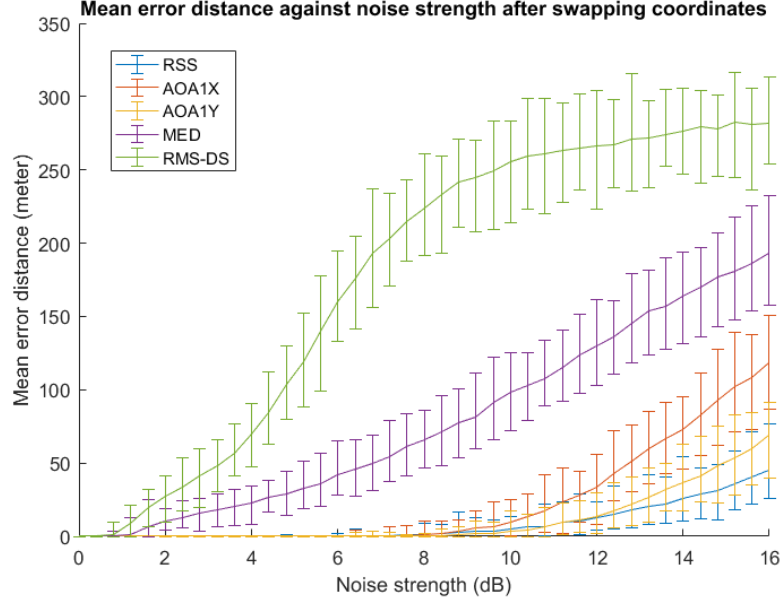


Figure 5-14: Mean error distances of relying on RSS, AOA1X, AOA1Y, MED or RMSDS against biases after reversing the coordinate system of the Hoxton scenario; the curves represent the average of 100 simulations, and the error bar represents the minimum and maximum.

The width of adjustment window of the DTW specifies the number of neighbour samples to be considered along the warping path which may affect the fingerprint matching accuracy. Therefore, an investigation is carried out by increasing the width of adjustment window from 1 to 6, which is applied to RSSs in the presence of 8dB AWGN. As can be seen from Table 5.3, the average accuracies are very similar over 100 simulations. The best accuracy, 96.84%, is obtained when the width of adjustment window equals three. Since when width of adjustment window equals one the accuracy still remains close to the optimum, it can be concluded that the mismatch between corrupted fingerprints and the authentic fingerprint is usually no more than one sample.

Table 5.3: Average accuracies relying on RSS over 100 simulations when the width of adjustment window for the DTW algorithm increases from 1 to 6. (8dB AWGN is applied to the simulated RSSs)

Width of adjustment window	1	2	3	4	5	6
Accuracy (%)	96.50	96.75	96.84	95.73	96.24	95.81

Generally, it can be concluded that MED or RMSDS measurement alone is not capable of source localization. Variations of RSS and the most significant AOA collected by a UAV have the potential of reliably revealing source locations. The width of

adjustment window in terms of DTW is less significant for the successful matching of fingerprints in this localization approach. However, there are yet doubts on whether any UAV path in the environment is capable of telling source locations, and what kind of route strategy determines the best localization accuracy. These are considered in the next section.

### 5.2.3 Route selection

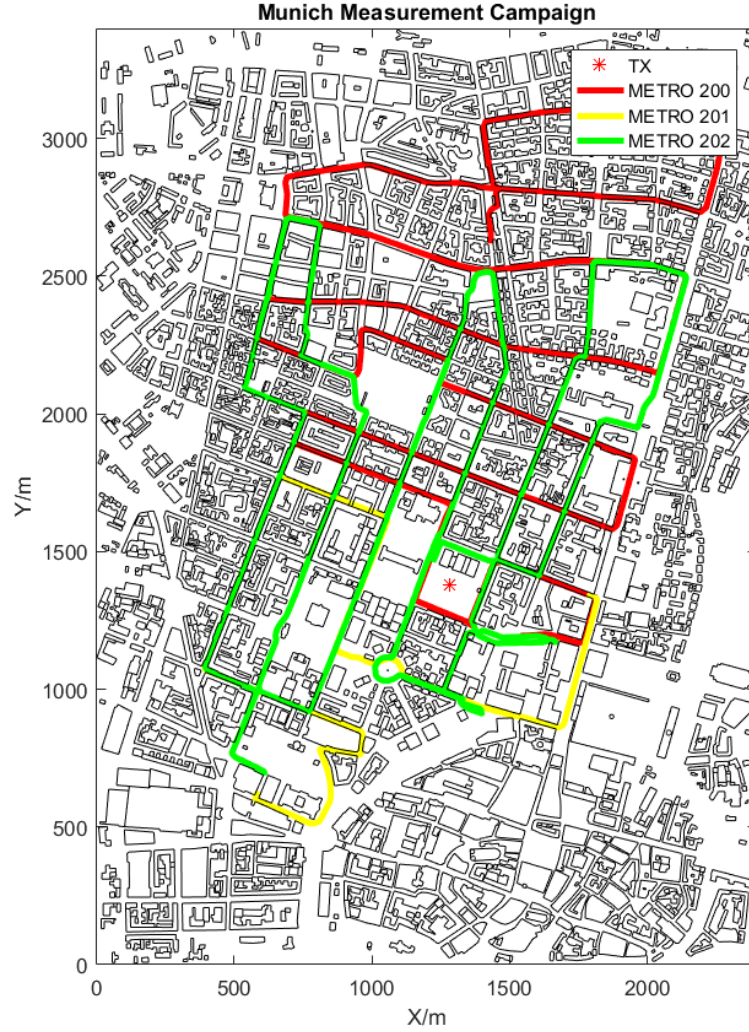


Figure 5-15: Measurement campaigns in Munich downtown area with three routes, named as ‘METRO 200’, ‘METRO 201’ and ‘METRO 202’.

We have been using the COST 231 action measurements from Mannesmann Mobilfunk GmbH [36] to validate our 2.5D ray launching model in Chapter 3. In this

dataset, there are three different measurement routes available with an approximately constant sampling rate as shown in Figure 5-15. During measurement, the equipment was held 1.5m above ground and the separation distance between samples is 14m on average. Square pixels of length 10m have been allocated across the map which are adequate to simulate the observations.

We could simply assume the UAV is flying 1.5m above ground while getting path loss measurements as shown in Figure 5-16. Thus, the source localization algorithm can be applied to these three routes to test their performance.

As stated in Table 5.1, there are 970, 355, 1031 samples along routes ‘METRO 200’, ‘METRO 201’ and ‘METRO202’, respectively. We were able to simulate  $28 \times 22 = 616$  emitter candidates in a same configuration as the real one which transmits a 1W narrow-band signal at 947MHz through omni-directional antennas. These emitters are assigned in a fashion of grids with 100m uniform separation.

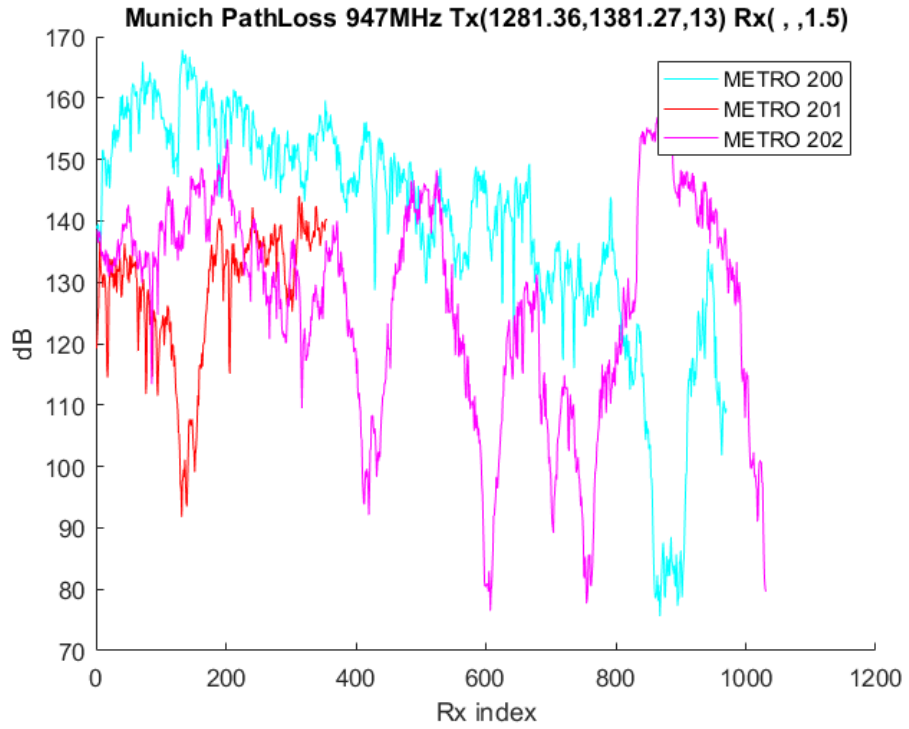


Figure 5-16: Pathloss trends along the three measurement routes on the Munich scenario.

Using the ray launching model, we obtained the RSS distributions with respect to all emitter candidate locations. Figure 5-17 shows the estimated RSS heat map given the source location at (1281.36m, 1381.27m). The trajectories of the measurement campaign are plotted against the simulated heat map as shown in Figure 5-18.

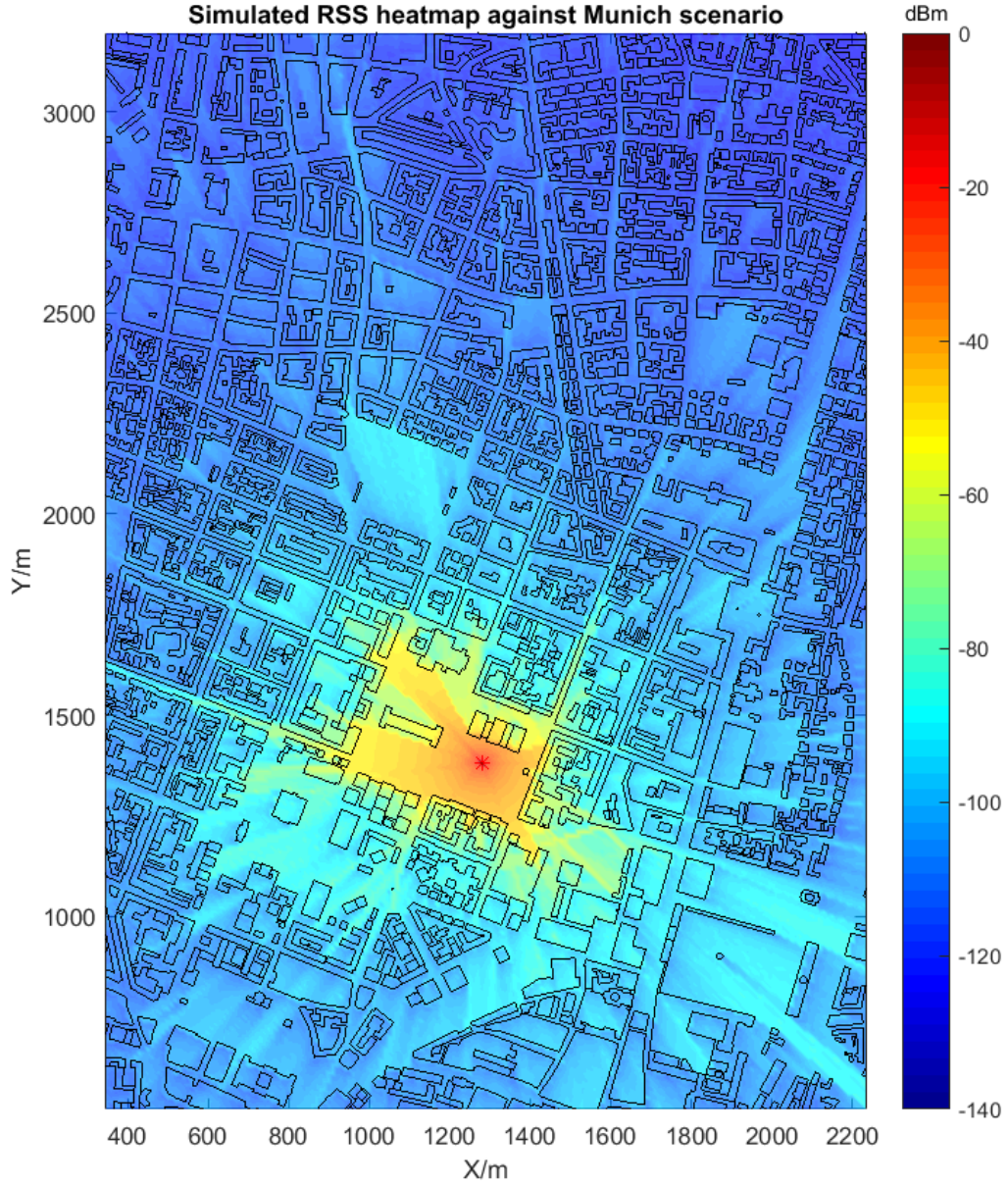


Figure 5-17: Simulated RSS heat map given a 947MHz radio source transmitting 1 W power at (1281.36 m, 1381.27 m) on the Munich scenario.

The closest grid point at (1300 m, 1400 m) is expected to be the best match. Comparisons between simulated RSS trends when the emitter is at (1300 m, 1400 m) and the measured ones can be found in Figures 5-19, 5-20, 5-21.



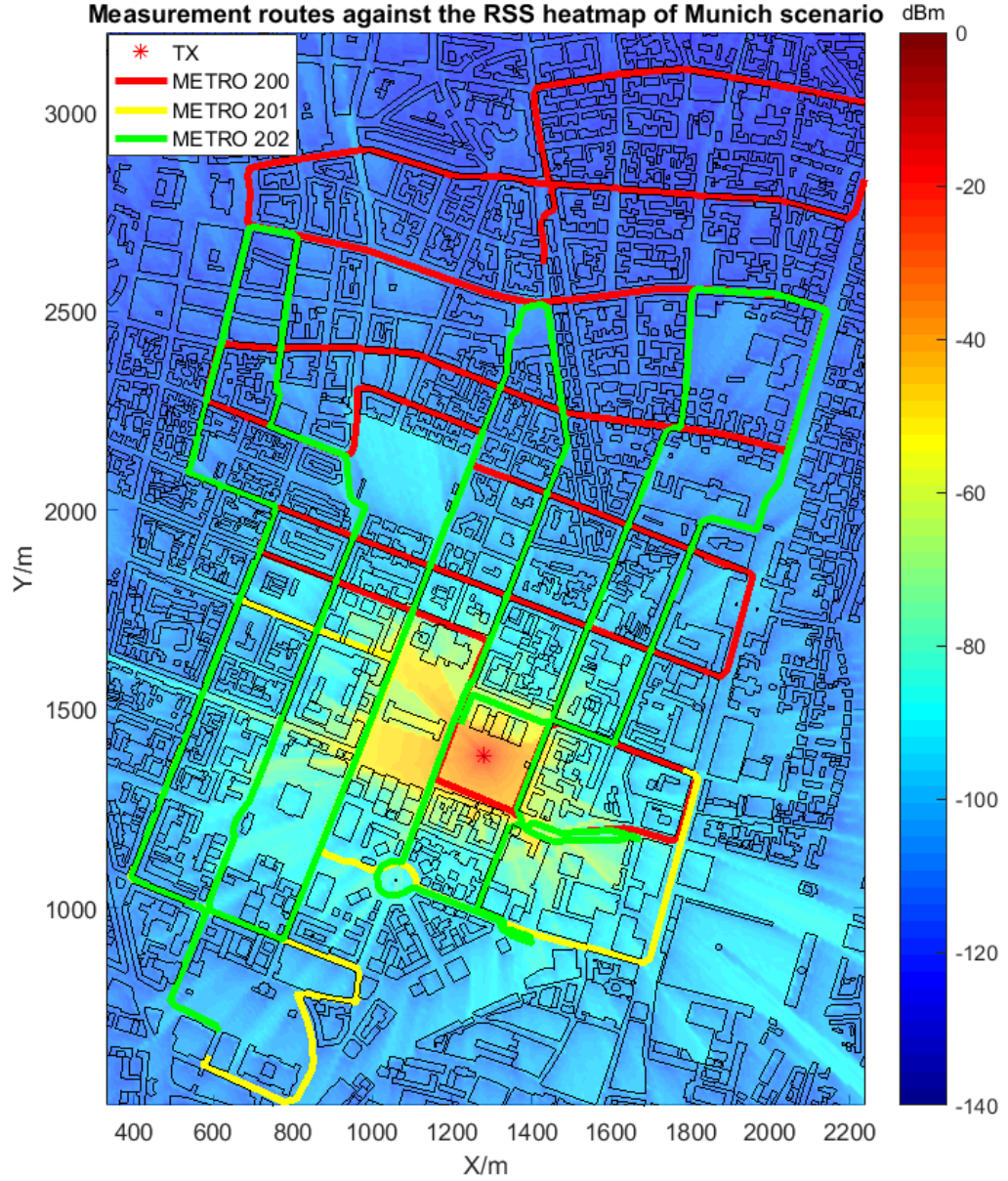


Figure 5-18: Measurement routes against RSS heat map given a 947 MHz radio source transmitting 1 W power at (1281.36 m, 1381.27 m) on the Munich scenario.

As has been shown in section 3.5, the agreement between measurements and simulations is good (Mean Absolute Error is between 5.7 dB and 7.7 dB) and the correlation coefficient between simulations and measurements is high (over 0.8725 for any measurement route). Importantly here the deep fades are well represented which gives optimism that the spatial frequencies can be reliably predicted.

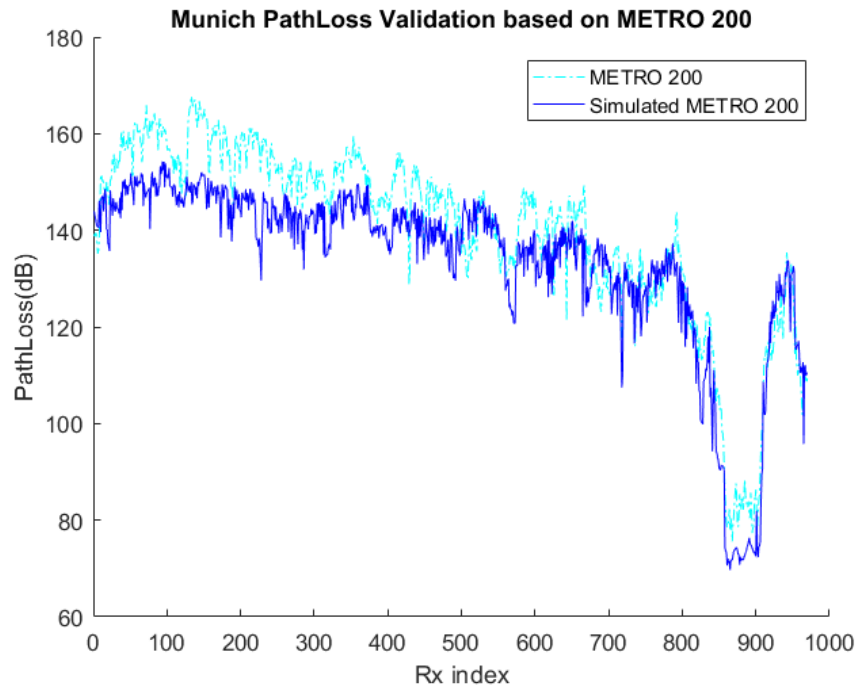


Figure 5-19: Simulated RSSs along ‘METRO 200’ compared to measurements.

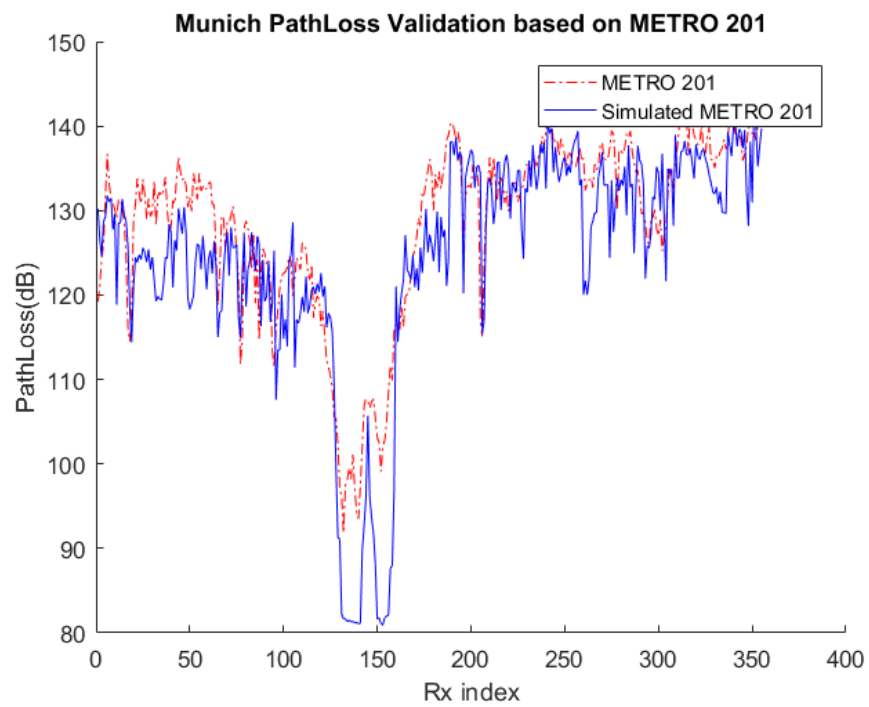


Figure 5-20: Simulated RSSs along ‘METRO 201’ compared to measurements.

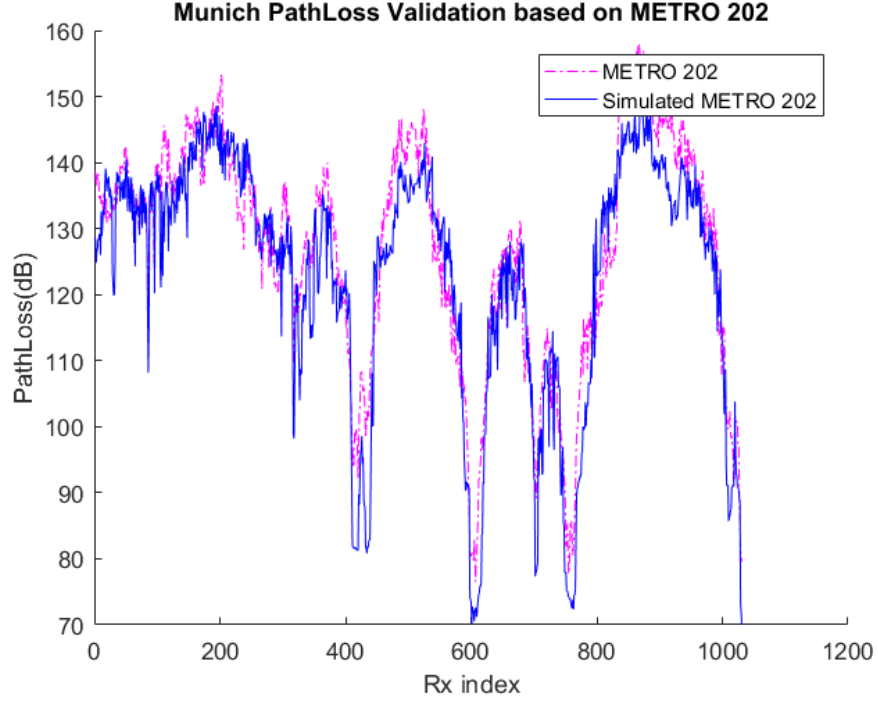


Figure 5-21: Simulated RSSs along ‘METRO 202’ compared to measurements.

Following the source localization procedures, we established fingerprint databases based on simulated RSSs along the three different routes. Each series of measurements can be regarded as an entry to derive a UAV observed fingerprint. This fingerprint is subsequently exposed to increasing levels of noise strength and bias, and searched against the database using the proposed matching algorithm.

Figure 5-22 exhibits the localization error performance against noise strength and bias where darker colours represent a smaller error and brighter colours suggest a larger error. The dark blue indicates ‘spot on’ in which the closet source location is found. Light blue indicates an error of less than 150 m, i.e., a neighbour of grid (1300 m, 1400 m) is located. The yellow and red dots are worse estimations, being at least 200 m away.

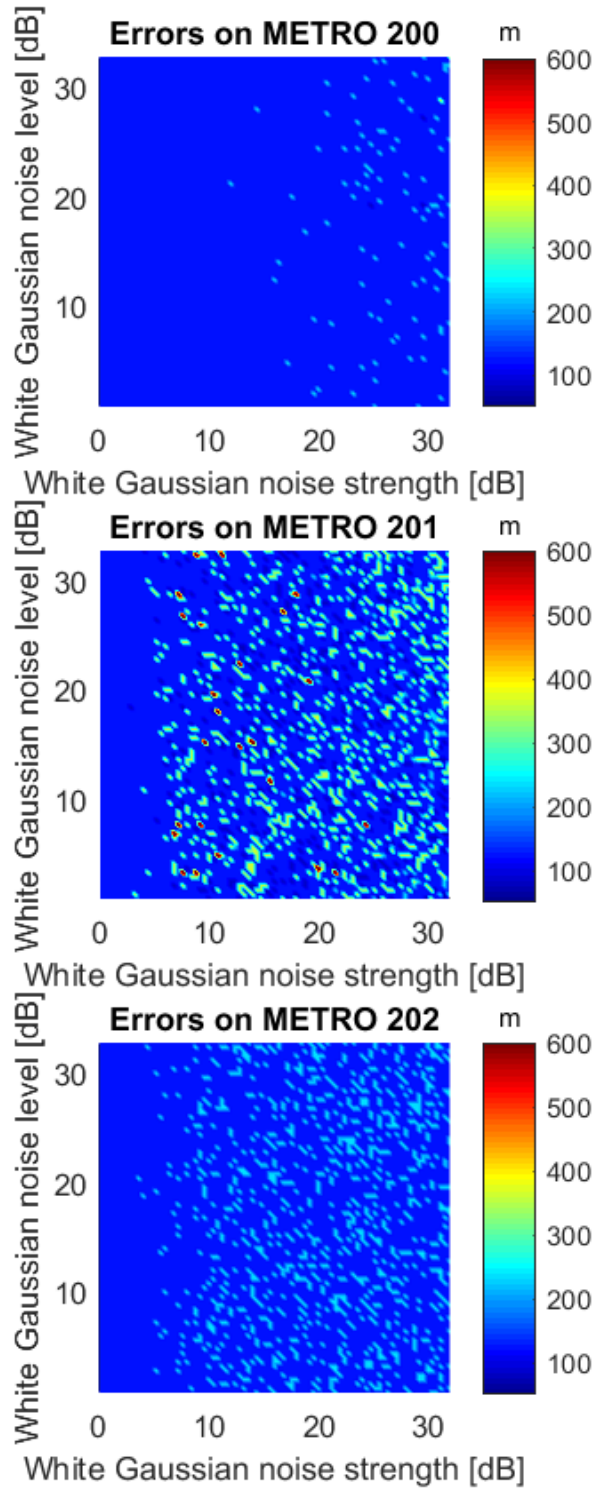


Figure 5-22: Localization errors for the three routes against AWGN strength (standard deviation) and bias (mean) where brighter colours indicate large errors and darker colours indicate small errors.



There are several interesting characteristics seen in the error distributions. Firstly, as long as the AWGN strength added to the RSS observation is less than 5dB a best match is mostly returned along any of the three routes. No matter how much bias there is, the minimum error always holds. This discovery confirms that the DTW is working very well in realizing similar shapes between two fingerprints while neglecting the absolute value difference. Since noise and bias added to the fingerprint may shift amplitudes and phases of the frequency components, the envelope of these low frequency components, i.e., the fingerprint, sees distortion which, nevertheless, can be mitigated by DTW algorithm.

The two worse routes, ‘METRO 201’ and ‘METRO 202’, remain relatively robust to a 5dB noise strength and any large bias. This suggests the source localization algorithm may be generalized even though the UAV path cannot be freely chosen.

It is clear that the ‘METRO201’ is the worst route among the three. As the noise strength increases, absurd location estimations are witnessed. The cause of these errors is mainly an insufficient number of samples. If the sample separations are identical, a lack of samples indicates a limited travel distance. Thus, it is problematic to extract spatial frequency components for short distances. In comparison to ‘METRO 201’, the other two routes have double of the number of samples. As a result, ‘METRO 200’ and ‘METRO 201’ can generally make an estimation no worse than 200m in error given an extremely noisy and biased channel. Hence, these results indicate that a longer UAV route which provides sufficient spatial frequency components is preferable.

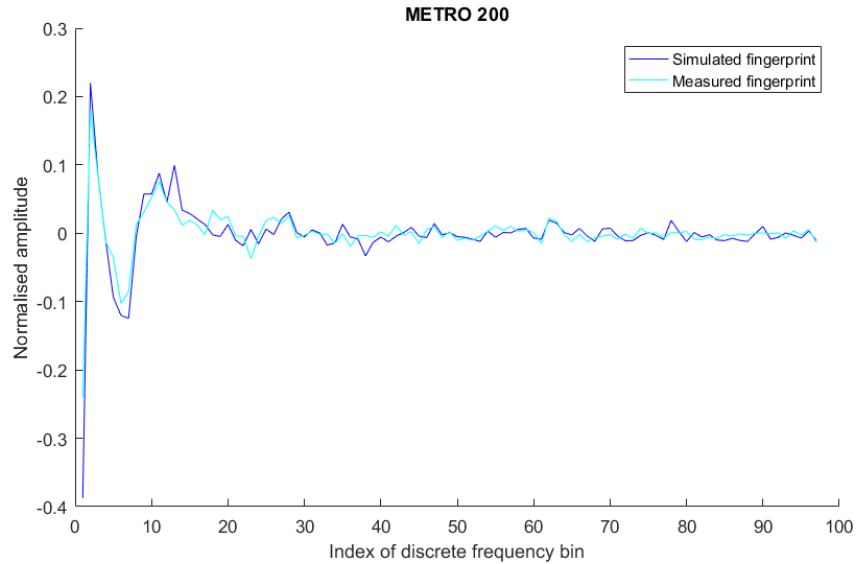


Figure 5-23: Light blue envelope represents measured fingerprint of route ‘METRO 200’; blue one is the corresponding best match found among the fingerprint database.

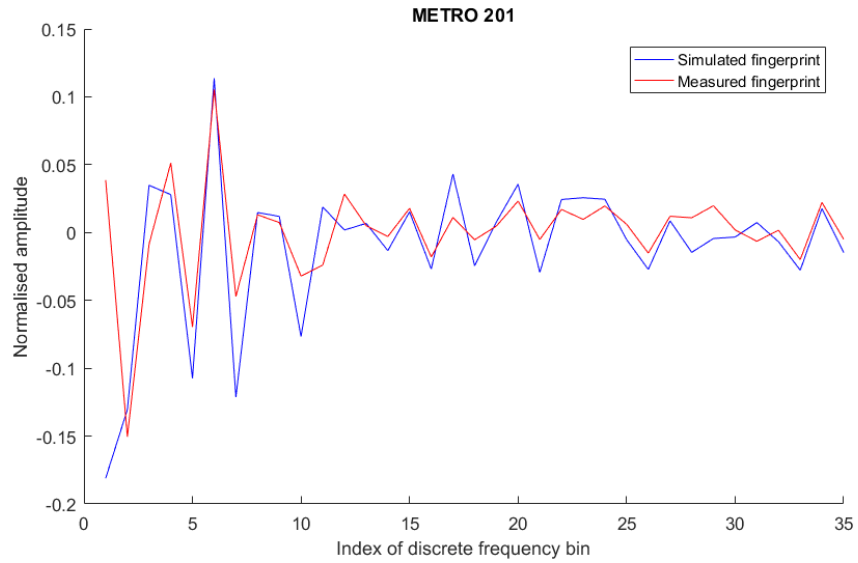


Figure 5-24: Red envelope represents measured fingerprint of route ‘METRO 201’; blue one is the corresponding best match found among the fingerprint database.

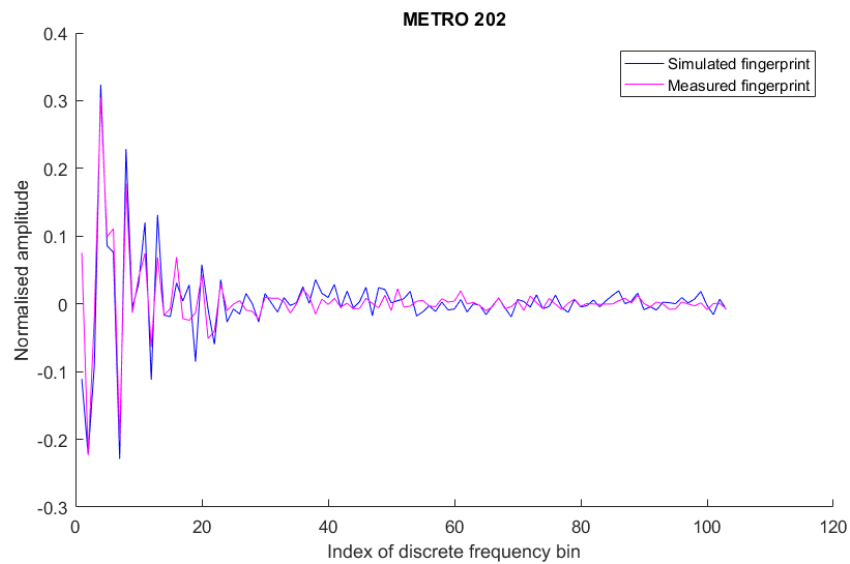


Figure 5-25: Magenta envelope represents measured fingerprint of route ‘METRO 202’; blue one is the corresponding best match found among the fingerprint database.

Lastly, ‘METRO 200’ seems superior to ‘METRO 202’ although they are of similar length and both wind through street canyons. We suppose it is the general moving direction that differentiates them. If a closer look is taken at their fingerprint envelopes in Figures 5-23, 5-24 and 5-25, it is easy to see that the fingerprint of ‘METRO 200’ in its DFT has one significant low frequency component at the third frequency bin,

as well as a series of high strengths at mid-frequency range. Whereas, the signature of ‘METRO 202’ has considerable ripple in the low frequency region which makes it vulnerable to misalignment with the target fingerprint using the DTW algorithm.

On the map of Figure 5-15, ‘METRO 200’ which is marked in red and starting from north part of the map is gradually moving towards south, i.e., the real source position. This approaching direction is essentially generating a slowly rising trend of RSS which gives rise to the significant low frequency impulse seen in its DFT. However, ‘METRO 202’ marked in green repeatedly approaches and departs from the source which is likely to generate a series of less outstanding spatial variations. It is highly likely such ripple in the low frequency components are mismatched if DTW does not determine a perfect alignment. Therefore, a UAV route that is definitely approaching or leaving the source location is preferred. Notice that the source location is unknown *a priori*. As a result, a proper route strategy should be expressed as generally moving towards one direction without weaving back and forth.

In brief, RSS is the most reliable primitive against various kinds of interferences in terms of constructing source location fingerprints. A preferable observing route should generally head towards one direction while having as longer extension as possible.

### 5.3 Summary

This chapter details the development of the radio source localization scheme. A source location fingerprint is defined relying on field measurements from a UAV. After transformation of the measured data, a database is constructed containing fingerprints of all source candidate locations. A DTW-based matching algorithm is used to find the nearest source location. This method has been tested exposed to measurement errors, deviations, AWGNs and different UAV routes. Outstanding accuracy and universality of use are validated.

Although a longer measurement route is preferable for the case studies in this chapter, this may not be true if the route extends too long. Since the lower frequency components which represent spatial variations over a relatively longer distance are more capable of determining source locations, they would be combined with more measurements to make higher frequency ripples along a much longer route. Whether the spatial frequency components of a much longer route are still trustworthy to determine source locations, and how long a route should be to offer the best localization performance are not clear, and could be part of further work.

## Chapter 6

# Enhancement and program optimization

The proposed navigation and source localization approach largely relies on the development of a propagation model and localization algorithms. Optimization of the system is a key step to guarantee the success. The fingerprints of locations are inherently static as long as the opportunistic signals and propagation environment remain unchanged. In Chapter 4, it has been shown that small changes to the environment do not result in significant errors. Thus, we attempt to share the static fingerprint database with users such that they are able to navigate locally. In this chapter, a sequence-based step detection scheme which aims to minimize the communication between user and server is proposed to assist the navigation system during off-line periods.

For the purpose of building channel fingerprint databases, the 2.5D ray launching model developed in this work has high requirements for both accuracy and efficiency. Matlab is an extraordinary platform for scientific research and development. Not only does it provide easy handling of matrix and structure array, but comprehensive embedded functions on visualization, data manipulation as well as portability to other software. In some instances, there is a significant performance overhead. The adoption of Matlab allows for rapid development and high accuracy but not always the highest speed. In this chapter, the methods adopted to optimize the Matlab master program are expanded in detail.

The use of GPU based multi-threading computing is deemed an effective means for acceleration. In this chapter, we briefly explain the parallel computing compatibility of our program and introduce the advantages as well as limitations of using GPU. The implementation of this accelerator has been done with the help of CUDA and the speed up performance has been evaluated. Eventually, above acceleration and optimization

methods make accurate and rapid propagation modelling a realistic prospective.

## 6.1 Sequence-based navigation scheme

One difficulty of fingerprinting techniques that restrict the precision of localization stems from the reliability of the fingerprint data. Unreliable fingerprints which are likely to result in poorly estimated locations may come from interference, fast fading or path obstructions. As the mobile user usually proceeds along a continuous path, a consideration might be if previous footsteps could be exploited to correct the navigation solution along the path by avoiding odd location estimations.

This may help with continuity since a mobile device occasionally suffers from lost signals in densely built areas. Thus, we propose to navigate the user with as little communication between the mobile device and the server as possible. Given the localization algorithm and preliminary fingerprint database, our proposed navigation system adopts a built-in sequence detection algorithm adapted from an indoor sequence-based approach [129, 130].

Based on the assumption a mobile user will go along a continuous path, a package of neighbour fingerprints is passed to the user among which the next step is expected. Since 5 m or 10 m pixel length is usually applied, a 12-neighbour detection (see Figure 6-1) is used to find the next step in a short interval, e.g. 2 seconds. During 2 seconds, a pedestrian user can be assumed to move for no more than 10 m. Consequently, 12-neighbour pixels are promising to identify a position change of up to 15.6 m.

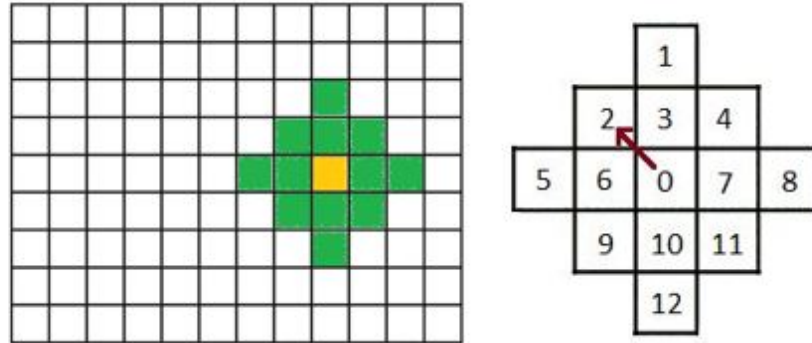


Figure 6-1: An illustration of the sequence-based navigation method when position number 2 is the best anticipated next step.

On the mobile device, a simple and reliable algorithm, *K*-Nearest Neighbours, is adopted to calculate the closest next step. As is shown in Figure 6-1, if sequence

number 2 gives the minimum difference using KNN it is decided as the next step. If number 0 is the best match, the user is assumed to stay stationary.

The package sent to the user contains all the fingerprint primitives of 12 candidate locations which may still be too many, and some of them having small weight after training. Therefore, during the off-line phase principal component analysis is applied to the database of fingerprints to determine the most significant primitives. For example, three among the RSS, MED, RMSDS, AOA1X, AOA1Y etc. may be chosen whose combined weights matter the most. If RSS, AOA1X and AOA1Y are taken, the step index  $i$  is exhaustively searched from 1 to 12 to find the minimum of the expression below

$$\{w_1 \cdot (RSS - RSS_i)^2 + w_2 \cdot (AOA1X - AOA1X_i)^2 + w_3 \cdot (AOA1Y - AOA1Y_i)^2\}$$

where  $w_j$  represents the assigned weight of  $j$ -th selected primitive. To prevent the accumulation of errors over time, periodically it may be necessary to send a complete observed fingerprint to the server to correct its current location.

Following this approach, a sequence-based database can be obtained from the original fingerprint database for every pixel. Prior to the starting of navigation process, a package of sequence-based database is delivered to the mobile device, so that the next few steps of the user can be automatically decided by monitoring the observed fingerprint against the sequence-based database which is stored locally.

Hence, the built-in navigation algorithm is able to guide the user by tracing an optimized next step. This method makes most of the localization effort off-line which effectively reduces the required communication between users and server. This greatly improves the practicality of the navigation system.

## 6.2 Matlab

The master program of the 2.5D ray launching model has been developed in Matlab 2016b. The inputs to this program include initiation of the source ray wavefront, layout of the environment and specifications of the model, such as radius of the reception sphere, reflection limit, truncation range of RSS etc. The ray launching mechanisms described in Chapter 3 work together to produce channel estimations. Dependant variables RSS, TOA, AOA with regard to locations are the outputs which are stored in a cell array. A flowchart of this master program is shown in Figure 6-2.

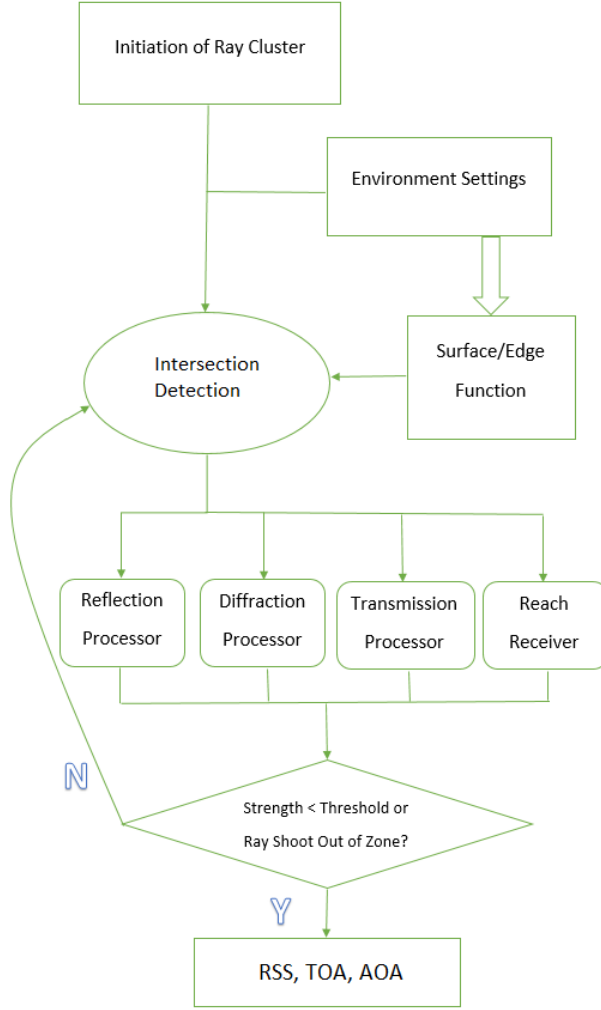


Figure 6-2: A flowchart of the master program of the 2.5D ray launching model.

A primary unit of the ray launching engine detects the intersection situation between a vector, i.e., a ray, and a line segment, i.e., a building surface. Depending on the type of intersection, a reflection processor, diffraction processor or transmission processor will determine the propagation behaviour onwards. A critical bottleneck that restricts the performance of CPU-based program is the tremendous number of *for* loops, which analyse intersections between one among the equally-radiated rays and one building surface at a time. In this work, it takes even longer if a fully-connected mesh of transmitter and receiver relationships is the final objective. The pseudo code of the intersection detection unit can be written as follows;

---

**Algorithm 1:** Intersection detection function for determining the ray propagation behaviour

---

```

1 Intersection Detection;
   Input : Virtual source  $O(x_o, y_o)$ ; ray  $\vec{r}$ ; surface  $(x_a, y_a) \rightarrow (x_b, y_b)$ 
   Output:  $[flag, distance]$ 
2 if  $intersection = true$  then
3   | if  $intersection\{type\} = edge$  then
4   |   | return  $[flag, distance] = Diffraction( )$ ;
5   | else
6   |   | return  $[flag, distance] = Reflection( )$ ;
7   |   | go to  $Transmission(\vec{r}, distance)$ ;
8   | end
9 else
10  | return  $[flag, distance] = set(0)$ ;
11 end

```

---

For instance, if 720 rays with up to 7 reflections are shot and 34890 walls (which are from the Munich map occurred in the scenario), to calculate 100 emitter candidates and 54471 receiver pixels it takes about  $9.58 \times 10^{14}$  iterations of the intersection detection unit. A brute force calculation based on a 2.6GHz Intel Core i5 dual-core processor would take many month to compute.

To reduce the computation time, we have adopted quite a few means of acceleration. Firstly, we omit calculations which do not make sense in physics, e.g., rays shooting out of the region, termination of rays diffracted or dropped below a signal strength threshold. Secondly, we abandon built-in Matlab functions which have a significant overhead by using simplified math operations instead. For instance, the embedded *cross product* function is called many thousands of times. Since we actually only use a summation of scalar products to obtain what is needed, a lot of time can be saved.

Note that default unit of storage in Matlab is *double* which takes 64 bits. We do not require that much precision, so we assign *single* precision uniformly which halves the overall memory space. It also accelerates the calculation to a degree. More importantly, this barely allows us to run the program on the Munich scenario with 16GB RAM. At least 32GB RAM is recommended for source localization study on a larger scenario.



### Profile Summary

Generated 24-Aug-2017 13:04:20 using performance time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
<a href="#">EXPERIMENT2D_30_Paris</a>	1	1379.216 s	0.554 s	
<a href="#">SCAN31</a>	1	1374.163 s	127.681 s	
<a href="#">shoot_ray10</a>	720	1190.739 s	88.674 s	
<a href="#">MexFunction5448</a> (MEX-file)	223281	1077.346 s	1077.346 s	
<a href="#">RoofTopDiff3</a>	1	45.245 s	24.753 s	
<a href="#">reception_tree</a>	7745660	23.638 s	23.638 s	
<a href="#">reception3</a>	2858	20.460 s	20.460 s	
<a href="#">diffractivity</a>	5986	9.722 s	0.219 s	

Figure 6-3: An example of Matlab profiling report when calculating propagation paths on the Arc de Triomphe scenario.

Matlab itself has a profiler which analyses processing time spent on functions and syntaxes in full detail. An example of profiling the Arc de Triomphe scenario is shown in Figure 6-3. The decomposition of self time allows us to trace a real bottleneck of the program. Although ‘shoot ray10’ cost over 1190 seconds most of its processing time was due to its sub-function called ‘MexFunction5448’ (see the next section for the configuration of this CUDA function). As can be seen, this function took majority of the total time and is being called for thousands of iterations which should be the focus if better performance is expected. A parallel computing solution to address the issue will be discussed in the next section.

Matlab provides powerful toolboxes on machine learning and data mining as well. The Artificial Neural Network has been detailed in Chapter 4 to aid a navigation solution. User customized ANN configurations, such as training algorithm, transfer functions, layers and neurons etc. can be explored. In Chapter 5, Dynamic Time Warping was applied to calculate fingerprint distances. A built-in DTW is available in Matlab but could be further accelerated.

## 6.3 GPU-based acceleration

As can be seen from the profiling results in Figure 6-3, most of the time is spent on hundreds of thousands of intersection detections. According to the literature [32], Graphics Processing Units (GPU) are equipped with mathematical operation functions in a parallel configuration which sees broader application in smart computation. It

has previously been noted that the intersection detections might benefit from GPU parallel computing since they possess remarkable features of parallelism. A significant characteristic during ray launching processing is that rays do not interfere along their own propagation paths. This means from departure to arrival, rays are being calculated independently which is advantageous for a GPU-based computing architecture. A general schematic diagram of GPU can be found in Figure 6-4. The most significant characteristic of this architecture is the large number of stream processors which operate in parallel. Thus, the large number of ray interactions can be spread across these many processors.

Hence, the intersection units has been reimplemented using Compute Unified Device Architecture (CUDA) developed by NVIDIA. One advantage of CUDA is the existing wide usage, and many libraries are available. It is an open architecture that allows user customization and low-level manipulation. Last but not least, NVIDIA graphic cards are readily available. The GPU used in this work is GeForce GTX 960M in a Dell XPS15 laptop having a CUDA compute capability of 5.0 (Maxwell). This laptop is equipped with an Intel i7 CPU with 16 GB RAM.

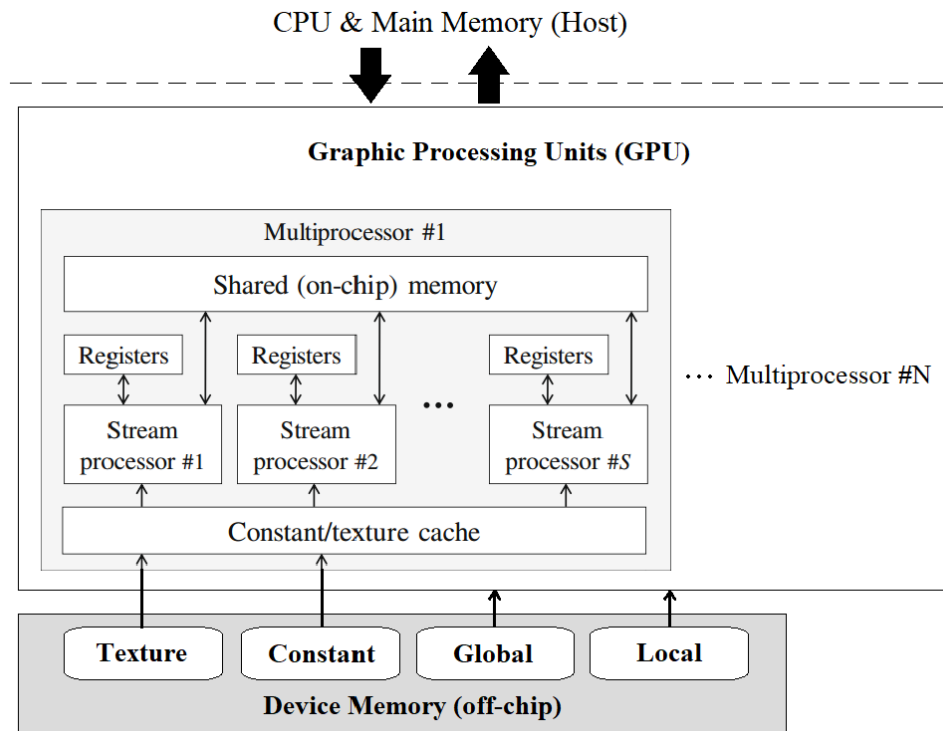


Figure 6-4: A schematic diagram on GPU memory accessibility and multiprocessor configuration.

The intersection detection functions have been implemented in Visual Studio 2015 using the CUDA 8.0 toolkit. This time the intersection calculations are now assigned in parallel to the GPU math units. Reading a ray and a surface coordinates from the CPU memory, the GPU kernels ideally complete the intersection concurrently and return the result back to host. Ideally, all ray-surface intersections can be dealt with in a single loop.

In reality, however, one of the bottlenecks using GPU lies in the expensive communication between host and device. Memory allocation must be explicitly performed for a CUDA thread to operate normally. The memory copy transfer rate between CPU and GPU is relatively limited meaning that transferred data should be minimized where possible. Moreover, threads are likely to be pipelined depending on the allocation of threads per block and blocks per grid in terms of multiprocessor usage. As a result, in common with all parallel computing approaches, the lowest throughput rate determines the overall speed.

Table 6.1: An analysis of processing time with respect to number of walls and CPU or GPU based architecture

Configuration		Time(s)		
Number of surfaces	Architecture	Total	Intersection	Ratio(%)
S=25	CPU	37.365	4.432	11.86
	GPU	178.686	137.701	77.06
S=1752	CPU	196.784	118.552	60.24
	GPU	219.121	170.151	77.65
S=5728	CPU	474.501	280.437	59.10
	GPU	224.511	122.201	54.43
S=34890	CPU	4452.890	3618.415	81.26
	GPU	1094.517	396.038	36.18

After testing four scenarios of different complexity, processing time results using CPU-based or GPU-based intersection detection when calculating one source are displayed in Table 6.1. It can be seen that when there are fewer walls, GPU-based performance is rather poor (in fact, slower than CPU only), most likely due to expensive data communication between host and GPU. As the number of walls increases, intersection detection processing time goes up for both CPU or GPU. However, CPU-based intersection takes up a much larger proportion of the total time and becomes a significant burden as can be more easily seen in Figure 6-5. GPU-based intersection units are more capable of dealing with greater number of walls, which is typically the case in

real-world urban environments. In the Munich scenario with 34890 walls, the GPU-based intersection detection function (which represents  $\approx 36\%$  of the total time) takes one-seventh of the CPU processing time. Generally, GPU enables an overall three-fold increase in computational performance in the simulation of typical urban environments.

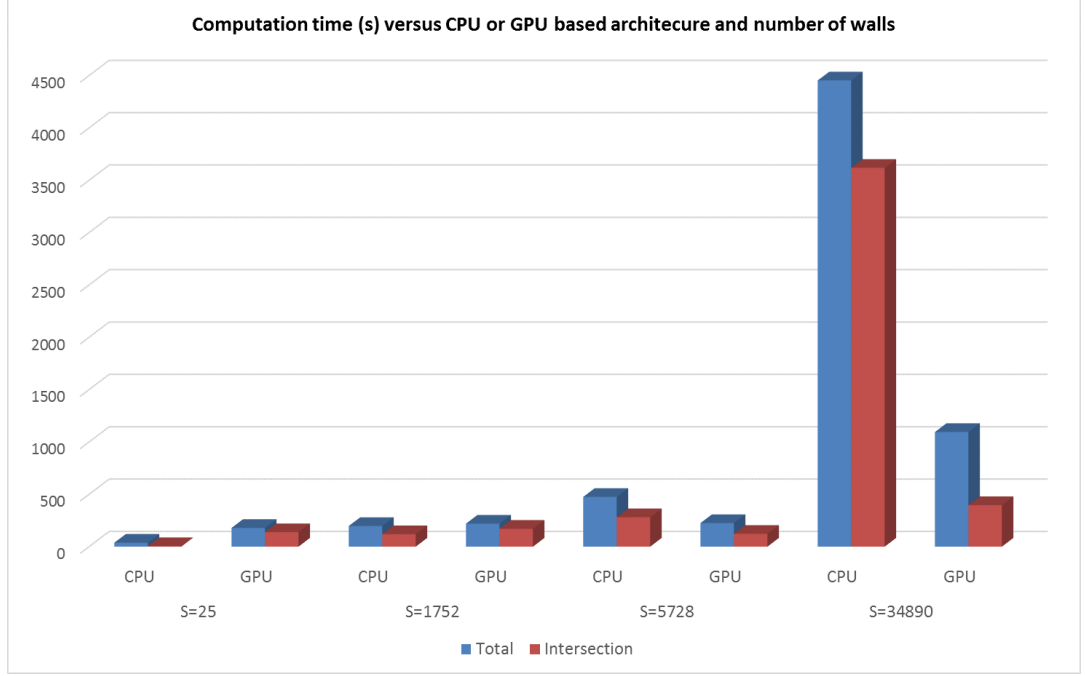


Figure 6-5: Computation time (s) using CPU/GPU on scenarios of different complexity, where blue bar represents total processing time; red bar indicates time spent on the intersection detection function; 'S' is the number of surfaces presented in the environment.

The Visual Profiler, also developed by NVIDIA, is a popular tool in evaluating CUDA performance. With the help of the Visual Profiler, Figure 6-6 shows a chart of six iterations of the intersection detection unit using GPU. Obviously, most of the time is spent on CUDA overheads in which the first-time initiation took up the greatest proportion. Thankfully, if an intersection unit is called thousands of times it only requires to set up once. A zoomed-in version of one iteration is displayed in Figure 6-7. We notice that the computation in threads, marked with a blue bar, takes much less than 'cudaMalloc' or 'cudaMemcpy' shown in brown. This implies the computation bottleneck is indeed the communication between host and device. Thus, the acceleration is yet far from completely optimized even though it is three times faster than pure CPU-based architecture.

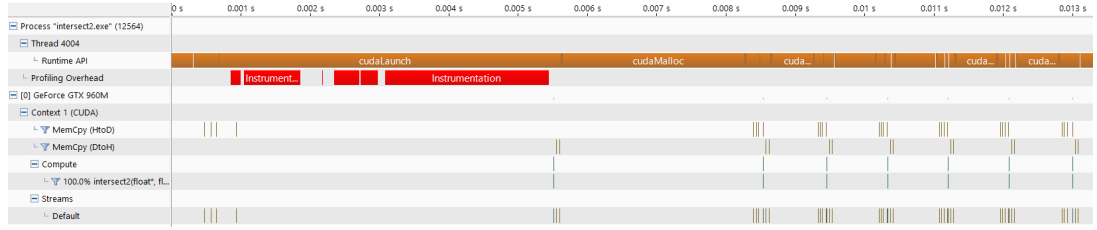


Figure 6-6: Profiling results on six iterations of the intersection function using the Visual Profiler.



Figure 6-7: Profiling results on one loop of the intersection function using the Visual Profiler.

A detailed GPU usage report can be found in Figure 6-8. The  $Compute/Memcpy = 0.418$  indicates a less than 1% ratio between thread computing and host-device communication, which ideally needs to be larger for optimal computation. An ideal situation is that threads are concentrating on local execution with little communication to the host. This means that computation flow is in parallel fashion and fewer threads are remaining idle for host-device communication. It is noted that GPU has vast potential for further acceleration and could be part of future work.

Several efforts have been made to raise the GPU usage. The 'Threads per Block'=1024 and 'Blocks Dimensions'=[1024, 1024, 64] were allocated from trial and errors method to determine the optimal configuration. These parameters determine how threads are configured in the CUDA parallel architecture. Furthermore, a 'cudaFree(0)' instruction is inserted before every iteration because it is the most swift procedure for waking-up the GPU compared to directly starting memory allocation. This change alone shortens the thread period by approximately 10%.

[0] GeForce GTX 960M	
GPU UUID	GPU-a602b6fd-8b81-bb32-2...
▼ Duration	
Session	15.876 ms (15,875,932 ns)
Compute/Memcpy	0.418
▼ Overlap	
Kernel/Memcpy	0%
▼ Attributes	
Compute Capability	5.0
▼ Maximums	
Threads per Block	1024
Threads per Multiprocessor	2048
Shared Memory per Block	48 KiB
Shared Memory per Multiprocessor	64 KiB
Registers per Block	65536
Registers per Multiprocessor	65536
Grid Dimensions	[ 2147483647, 65535, 65535 ]
Block Dimensions	[ 1024, 1024, 64 ]
Warps per Multiprocessor	64
Blocks per Multiprocessor	32
Single Precision FLOP/s	1.405 TeraFLOP/s
Double Precision FLOP/s	43.9 GigaFLOP/s
▼ Multiprocessor	
Multiprocessors	5
Clock Rate	1.098 GHz
Concurrent Kernel	true
Max IPC	6
Threads per Warp	32
▼ Memory	
Global Memory Bandwidth	80.16 GB/s
Global Memory Size	2 GiB
Constant Memory Size	64 KiB
L2 Cache Size	2 MiB
Memcpy Engines	1

Figure 6-8: An analysis of GPU usage based on the intersection function developed.

As has been stated in [104], the local memory access is extremely fast within each thread. However, if there is a race condition in accessing the same memory, delays will occur which would greatly impact the performance. Since each ray is being searched against all surfaces, regular assignment on the variables will result in surfaces competing to request a ray for operation. One of the best solutions for this problem is the constant memory which is globally visible across all threads of the GPU and easily accessible to locals. There are only 64KB of constant memory in common GPUs offering extremely high-speed reading, however, these cannot be written to. By placing the ray vector in constant memory, the intersection detection function is typically accelerated by over 30%.

Finally, the GPU-based intersection detection outputs are delivered back to the master program for subsequent calculations. This is implemented by *Mex*-function encapsulation. The processing results of summoning CUDA as well as thread allocation and execution are configured as a *Mex*-function to be returned to Matlab. Basically, the outputs are comprised of flags, which tell the intersection type of a ray, and distances,

which indicate travelled distances starting from a virtual source to the next intersection surface or edge. These are wrapped as a *Mex* array which is of the same format as Matlab variables.

## 6.4 Summary

In this chapter, a sequence-based navigation scheme to enable off-line usage has been proposed. This scheme can theoretically minimize the communication between user and server during navigation. The feasibility of using Matlab optimization means and GPU parallel computing techniques to accelerate our existing 2.5D ray launching model have been investigated. Simplifications of Matlab built-in functions and lower precision of variables was found to boost the processing speed. The use of parallel structures embedded within the model has been analysed. A replacement CUDA intersection detection function has been designed, tested and optimized. Compared to absolute CPU computation, the GPU-based intersection detection function runs seven times faster in large urban environments containing tens of thousands of walls.

Overall, it takes less than 190 hours to construct the off-line database of 616 source candidates in the Munich scenario. The Matlab optimization and GPU-based acceleration successfully multiply the computation speed of our ray launching model. These optimizations make the proposed navigation and source localization systems more practical.

## Chapter 7

# Summary and future work

### 7.1 Summary

Navigation and radio source localization in difficult urban environments has drawn significant industrial, military and civil engineering interest recently. However, the facts are that traditional satellite and terrestrial navigation systems do not perform well in complex urban environments. Currently, there is not a robust pervasive source localization scheme. This work has investigated into the feasibility of utilizing a propagation modelling approach, in which mature channel characterization capability has been demonstrated, and then integrated into a mobile and radio source localization solution. A fingerprinting technique is adopted to aid the development of the mobile localization and source localization algorithms. Following data mining and pattern recognition strategies, we are able to demonstrate reliable and accurate localization algorithms which have been tested in different real-world scenarios.

In order to generate adequate channel characteristic ‘truth data’ for location training, we developed a dedicate 2.5D ray launching model suitable for urban propagation simulation. The effectiveness of this model has been validated through checking against field measurement results. For the Munich city center environment, an absolute mean error of less than 8dB and a standard deviation of less than 7dB was found between measurements and simulation results from thousands of RSS samples. The efficiency of the model has been remarkably improved by optimizing the master program in Matlab and by exploiting CUDA to handle the extremely time-consuming intersection detections. The OpenStreetMap project has allowed us to bring digital maps into our environment modelling which requires precise shape and coordinate information of buildings and trees.

The mobile localization algorithm has been shown to produce a mean error of



approximately 13m for a 700m×550m residential area with a maximum reflection limit of seven. This algorithm is shown to remain robust against various errors and perturbations. The location fingerprint is defined based on the simulated PDP and angular spreads, which generate RSS, MED, RMSDS and AOAs parameters, with respect to the mobile positions. The configuration of the ANN has been studied and experimented with to yield an optimal trained network. A sequence-based localization method has been proposed to minimize the communication between users and server during navigation.

The source localization algorithm has been shown to confidently locate an emitter to a pixel of 100m within a 2.4km×3.4km city center area even when exposed to a significant degree of noise and different types of error. This is made possible with the help of a UAV which is expected to collect low-resolution channel characteristic parameters along a specified route. These parameters are then transformed into a source location fingerprint which is searched against simulated fingerprints of source candidates. Dynamic Time Warping has proved effective in calculating the fingerprint similarities. The stability of this algorithm has been verified by corrupting the measured fingerprint in different ways, such as the introduction of winds, reduction of sampling rate, increase in AWGN level etc. We have also proposed a general route selection strategy to increase the accuracy and reliability.

This thesis has broadly reviewed propagation modelling and localization methods with a focus on their limitations. From this a novel urban navigation and source localization solution has been developed and tested. New ideas and state-of-the-art techniques have been applied to construct a comprehensive urban propagation model and a new fingerprinted-based localization method. A key step in this work has been the development of an efficient 2.5D ray launching engine, as well as exploitation of recent advances in data mining and machine learning.

The accuracy and reliability of this localization system have been tested and confirmed. The author has also managed to optimize its performance for potential practical applications. The body of this work has met the objectives and answered the questions posed in the Chapter 1.

## 7.2 Future work

Although this thesis has answered the research questions posed in Chapter 1, many new questions have arisen. Future work may include refinement and calibration of the 2.5D ray launching model, development of a 3D ray launching model, investigation of the localization through deep learning on the digital maps directly, source local-

ization with unknown number of emitters and unknown mobility, and further GPU accelerations.

Perhaps the most obvious area for further improvement is the 2.5D ray launching model. Currently, rays are always deemed as vertically polarized which is clearly not true in real-world propagation. Comprehensive polarization of the electromagnetic waves should be considered. Many cities are built on top of non-flat terrain which significantly changes the propagation environment. Our model needs to take into account the complex landscape, including the modelling of hills, valleys, lakes, cliffs etc. Moreover, further measurements and tests in different types of environments are required to further calibrate the model and validate the localization algorithms.

Although it has been shown that 2.5D models are capable of channel modelling in urban scenarios, 3D ray launching models, widely used in indoor propagation modelling, allow even more accurate estimations. This is especially true for central business areas where significant building height difference is seen. In these scenarios, the roof-top diffraction model used in this thesis may not be valid. Based on our preliminary 3D modelling experiments in Chapter 3, we would suggest that a complete 3D ray launching model is implemented to take advantage of more accelerations to reduce the computational cost. In addition, the opportunistic sources are likely to be moving in urban environments. This may drive the need for a 4D model, which regards the channel characterization as a time-dependent function. A further option may also be to track the source through the spatial Doppler frequencies generated.

In order to correlate channel characteristics with locations, we applied ANN to the location fingerprints which were obtained by propagation modelling for one receiver candidate at a time. A future attempt to shorten this pattern recognition phase might be to directly train using the digital map and opportunistic sources to determine location. The hypothesis here is that the ray launching step may not be required as an intermediate step. In this case, the entire problem of fingerprinting to determine location is passed to the ANN.

In the assumptions of the source localization algorithm, we only committed to locate a single static emitter at a known frequency. It is worth investigating the more general case of the localization of multiple emitters with different velocities. This would almost certainly require a more powerful propagation modelling tool as well as more advanced localization approaches.

As noted in Chapter 6, there are more tiers of parallelism that can be recognized in the ray launching model and localization systems. For instance, rays shoot, propagate and bounce independently; the reception detections for all reference points are independent; the distance calculations between observed fingerprint and fingerprints in

the database are processed independently etc. These parallel structures are all options for further GPU-based acceleration to lessen the computation time. A wider range of acceleration schemes are likely to produce even better performance.

# References

- [1] MathWorks create, train, and simulate shallow and deep learning neural network. <https://uk.mathworks.com/products/neural-network.html>. Accessed: 13 May 2017.
- [2] OpenStreetMap official website. <http://www.openstreetmap.org>. Accessed: 30 September 2015.
- [3] Overpass Turbo user interface. <https://overpass-turbo.eu>. Accessed: 30 September 2015.
- [4] M. A. Abid and S. Cherkaoui. Wireless technology agnostic real-time localization in urban areas. In *2011 IEEE 36th Conference on Local Computer Networks*, pages 727–733.
- [5] S. Adelipour, M. Hamdollahzadeh, and F. Behnia. Constrained optimization of sensors trajectories for moving source localization using TDOA and FDOA measurements. In *2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM)*, pages 200–204.
- [6] F. A. Agelet, F. P. Fontan, and A. Formella. Fast ray tracing for microcellular and indoor environments. *Magnetics, IEEE Transactions on*, 33(2):1484–1487, 1997.
- [7] F. A. Agelet, A. Formella, J. M. Hernando Rabanos, F. I. de Vicente, and F. P. Fontan. Efficient ray-tracing acceleration techniques for radio propagation modeling. *Vehicular Technology, IEEE Transactions on*, 49(6):2089–2104, 2000.
- [8] F. Aguado, F. P. Fontan, and A. Formella. Indoor and outdoor channel simulator based on ray tracing. In *Vehicular Technology Conference, 1997, IEEE 47th*, volume 3, pages 2065–2069 vol.3.

- [9] T. Alwajeeh, P. Combeau, R. Vauzelle, and A. Bounceur. A high-speed 2.5D ray-tracing propagation model for microcellular systems, application: Smart cities. In *EuCAP 2017*.
- [10] H. R. Anderson. A second generation 3-D ray-tracing model using rough surface scattering. In *Vehicular Technology Conference, 1996. Mobile Technology for the Human Race., IEEE 46th*, volume 1, pages 46–50 vol.1.
- [11] H. R. Anderson. A ray-tracing propagation model for digital broadcast systems in urban areas. *Broadcasting, IEEE Transactions on*, 39(3):309–317, 1993.
- [12] L. Azpilicueta, M. Rawat, K. Rawat, F. M. Ghannouchi, and F. Falcone. A ray launching neural network approach for radio wave propagation analysis in complex indoor environments. *Antennas and Propagation, IEEE Transactions on*, 62(5):2777–2786, 2014.
- [13] K. Benzair. Measurements and modelling of propagation losses through vegetation at 1-4 GHz. In *Antennas and Propagation, 1995., Ninth International Conference on (Conf. Publ. No. 407)*, volume 2, pages 54–59 vol.2.
- [14] I. Bisio, M. Cerruti, F. Lavagetto, M. Marchese, M. Pastorino, A. Randazzo, and A. Sciarrone. A trainingless Wi-Fi fingerprint positioning approach over mobile devices. *IEEE Antennas and Wireless Propagation Letters*, 13:832–835, 2014.
- [15] D. Bojanjac, R. Nad, and Sisul G. Ray tracing model of pedestrian urban zone. In *ELMAR, 2010 PROCEEDINGS*, pages 289–292.
- [16] M. A. Caceres, F. Penna, H. Wymeersch, and R. Garello. Hybrid cooperative positioning based on distributed belief propagation. *IEEE Journal on Selected Areas in Communications*, 2011.
- [17] K. A. Chamberlin and R. J. Luebbers. An evaluation of Longley-Rice and GTD propagation models. *Antennas and Propagation, IEEE Transactions on*, 30(6):1093–1098, 1982.
- [18] G. Chen, Y. Zhang, L. Xiao, J. Li, L. Zhou, and S. Zhou. Measurement-based RSS-multipath neural network indoor positioning technique. In *Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on*, pages 1–7.
- [19] C. Cheon, G. Liang, and H. L. Bertoni. Simulating radio channel statistics for different building environments. *IEEE Journal on Selected Areas in Communications*, 19(11):2191–2200, 2001.

- [20] S. Coco, A. Laudani, and L. Mazzurco. A novel 2-D ray tracing procedure for the localization of EM field sources in urban environment. *Magnetics, IEEE Transactions on*, 40(2):1132–1135, 2004.
- [21] A. Coluccia and F. Ricciato. Maximum likelihood trajectory estimation of a mobile node from RSS measurements. In *2012 9th Annual Conference on Wireless On-Demand Network Systems and Services (WONS)*, pages 151–158.
- [22] O. Corte, A. D. Gutierrez and J. M. Gomez. High-accuracy localization based on the dominant rays of ray-tracing over fingerprinting techniques. In *Antennas and Propagation Society International Symposium (APSURSI), 2012 IEEE*, pages 1–2.
- [23] M. Curry, B. Koala, M. Ciccotosto, and Y. Kuga. Indoor angle of arrival using wide-band frequency diversity with experimental results and EM propagation modeling. In *Antennas and Propagation for Wireless Communications, 2000 IEEE-APS Conference on*, pages 65–68.
- [24] N. Czink, E. Bonek, Yin Xuefeng, and B. Fleury. Cluster angular spreads in a MIMO indoor propagation environment. In *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, volume 1, pages 664–668.
- [25] V. Degli-Esposti. Ray tracing propagation modelling: Future prospects. In *Antennas and Propagation (EuCAP), 2014 8th European Conference on*, pages 2232–2232.
- [26] H. Demuth and M. Beale. *Neural Network Toolbox For Use with MATLAB*. The MathWorks, 2004.
- [27] J. Deygout. Multiple knife-edge diffraction of microwaves. *IEEE Transactions on Antennas and Propagation*, 14(4):480–489, 1966.
- [28] G. Durgin, N. Patwari, and T. S. Rappaport. An advanced 3D ray launching method for wireless propagation prediction. In *Vehicular Technology Conference, 1997, IEEE 47th*, volume 2, pages 785–789 vol.2.
- [29] L. Felsen and N. Marcuvitz. *Radiation and Scattering of Waves*. The IEEE series on Electromagnetic Wave Theory. John Wiley and Sons Inc., 1994.
- [30] Y. Feng, L. Guo, P. Wang, and Z. Liu. Efficient ray-tracing model for propagation prediction for microcellular wireless communication systems. In *Antennas*,

*Propagation and EM Theory (ISAPE), 2012 10th International Symposium on*, pages 432–435.

- [31] C. Fernandez-Prades, L. L. Presti, and E. Falletti. Satellite radio localization from GPS to GNSS and beyond: Novel technologies and applications for civil mass market. *Proceedings of the IEEE*, (1882 - 1904), 2011.
- [32] J. C. Fernndez-Gonzlez, M. L. Portugus, Y. A. Lpez, J. A. Lpez-Fernndez, and F. A. Andrs. GPU-based computational acceleration of phaseless algorithms for antenna characterization. In *EuCAP 2017*.
- [33] V. Ferrara and C. M. Ottavi. Ray-tracing and UTD analyses for microcellular communications in urban scene: a software methodology that uses 3-D GIS data of buildings. In *Remote Sensing and Data Fusion over Urban Areas, IEEE/ISPRS Joint Workshop 2001*, pages 255–259.
- [34] C. Ghobadi, P. R. Shepherd, and S. R. Pennock. 2D ray-tracing model for indoor radio propagation at millimetre frequencies, and the study of diversity techniques. *Microwaves, Antennas and Propagation, IEEE Proceedings*, 145(4):349–353, 1998.
- [35] K. J. Gladstone and J. P. McGeehan. Computer simulation of multipath fading in the land mobile radio environment. *Electronic Circuits and Systems, IEEE Proceedings G*, 127(6):323–330, 1980.
- [36] Mannesmann Mobilfunk GmbH. *Cost 231 - urban micro cell measurements and building data*. 1999.
- [37] N. Goddemeier, S. Rohde, and C. Wietfeld. Experimental validation of RSS driven UAV mobility behaviors in IEEE 802.11s networks. In *2012 IEEE Globecom Workshops*, pages 1550–1555.
- [38] J. Goldhirsh and W. Vogel. Roadside tree attenuation measurements at UHF for land mobile satellite systems. *IEEE Transactions on Antennas and Propagation*, 35(5):589–596, 1987.
- [39] N. C. Goncalves and L. M. Correia. A propagation model for urban microcellular systems at the UHF band. *IEEE Transactions on Vehicular Technology*, 49(4):1294–1302, 2000.
- [40] E. Greenberg and Levy P. Propagation aspects for RF fingerprinting at open areas over irregular terrain. In *EuCAP 2017*.

- [41] Y. Gu, L. T. Hsu, and S. Kamijo. GNSS/onboard inertial sensor integration with the aid of 3-D building map for lane-level vehicle self-localization in urban canyon. *IEEE Transactions on Vehicular Technology*, 65(6):4274–4287, 2016.
- [42] K. Guan, G. Li, D. He, et al. Spatial consistency of dominant components between ray-tracing and stochastic modeling in 3GPP high-speed train scenarios. In *EuCAP 2017*.
- [43] D. He, J. Yang, K. Guan, et al. Ray-tracing simulation and analysis of propagation for 3GPP high speed scenarios. In *EuCAP 2017*.
- [44] J. Honda and T. Otsuyama. Rapid computation algorithm for radio wave propagation characteristics on airport surface. In *Complex, Intelligent and Software Intensive Systems (CISIS), 2014 Eighth International Conference on*, pages 302–306.
- [45] C. C. Huang and H. N. Manh. RSS-based indoor positioning based on multi-dimensional kernel modeling and weighted average tracking. *IEEE Sensors Journal*, 16(9):3231–3245, 2016.
- [46] S. Hussain and C. Brennan. An intra-visibility matrix based environment pre-processing for efficient ray tracing. In *EuCAP 2017*.
- [47] T. Imai and T. Fujii. Propagation loss in multiple diffraction using ray-tracing. In *Antennas and Propagation Society International Symposium, 1997. IEEE., 1997 Digest*, volume 4, pages 2572–2575 vol.4.
- [48] M. Jacob, S. Priebe, A. Maltsev, A. Lomayev, V. Erceg, and T. Kurner. A ray tracing based stochastic human blockage model for the IEEE 802.11ad 60 GHz channel model. In *Antennas and Propagation (EuCAP), Proceedings of the 5th European Conference on*, pages 3084–3088.
- [49] M. Jacob, S. Schon, U. Weinbach, and T. Kurner. Ray tracing supported precision evaluation for GPS indoor positioning. In *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on*, pages 15–22.
- [50] S. Jiang and S. V. Georgakopoulos. Optimum wireless power transmission from air to lossy media. In *Wireless and Microwave Technology Conference (WAMICON), 2011 IEEE 12th Annual*, pages 1–6.
- [51] Y. Jin, N. O’Donoghue, and J. M. F. Moura. Position location by time reversal in communication networks. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3001–3004.



- [52] J. Karedal, F. Tufvesson, T. Abbas, O. Klemp, A. Paier, L. Bernado, and A. F. Molisch. Radio channel measurements at street intersections for vehicle-to-vehicle safety applications. In *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, pages 1–5.
- [53] A. O. Kaya, L. Greenstein, D. Chizhik, R. Valenzuela, and N. Moayeri. Emitter localization and visualization (ELVIS): A backward ray tracing algorithm for locating emitters. In *Information Sciences and Systems, 2007. CISS '07. 41st Annual Conference on*, pages 376–381.
- [54] Joseph B. Keller. Geometrical theory of diffraction. *Journal of the Optical Society of America*, 52(2):116–130, 1962.
- [55] J. M. Kelner, C. Ziolkowski, L. Nowosielski, and M. Wnuk. Localization of emission source in urban environment based on the Doppler effect. In *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, pages 1–5.
- [56] E. Keogh and Ratanamahatana C. A. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 2004.
- [57] Y. Jun Y. Chong Kim, J. H. Yoon and H. J. Hong. Millimeter-wave delay spread measurement and simulation at LoS urban low-rise environments. In *Information and Communication Technology Convergence (ICTC), 2015 International Conference on*, pages 1194–1196.
- [58] C. Kleist. *Time Series Data Mining Methods: A Review*. Master’s thesis, School of Business and Economics, Humboldt University of Berlin, 2015.
- [59] T. Kopacz, S. Schiebl, and D. Heberling. Investigating the overestimation of base station exposure in urban environments due to assumption of free space propagation. In *EuCAP 2017*.
- [60] R. G. Kouyoumjian and P. H. Pathak. A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface. *Proceedings of the IEEE*, 62(11):1448–1461, 1974.
- [61] Y. Kurdi and L. de Haro. DOA measurements on indoor channel based on music and MDL processing. In *Antennas and Propagation Society International Symposium, 2008. AP-S 2008. IEEE*, pages 1–4.
- [62] T. Kurner, D. J. Cichon, and W. Wiesbeck. Concepts and results for 3D digital terrain-based wave propagation models: an overview. *IEEE Journal on Selected Areas in Communications*, 11(7):1002–1012, 1993.

- [63] M. Laaraiedh, N. Amiot, and B. Uguen. Efficient ray tracing tool for UWB propagation and localization modeling. In *Antennas and Propagation (EuCAP), 2013 7th European Conference on*, pages 2307–2311.
- [64] Z. Lai. *The Development of an Intelligent Ray Launching Algorithm for Wireless Network Planning*. PhD thesis, University of Bedfordshire, 2010.
- [65] Z. Lai, H. Song, P. Wang, H. Mu, L. Wu, and J. Zhang. Implementation and validation of a 2.5D intelligent ray launching algorithm for large urban scenarios. In *2012 6th European Conference on Antennas and Propagation (EuCAP)*, pages 2396–2400.
- [66] M. C. Lawton and J. P. McGeehan. The application of GTD and ray launching techniques to channel modelling for cordless radio systems. In *Vehicular Technology Conference, 1992, IEEE 42nd*, pages 125–130 vol.1.
- [67] J. F. Li, J. Wagen and E. Lachat. Propagation over rooftop and in the horizontal plane for small and micro-cell coverage predictions. In *Vehicular Technology Conference, 1997, IEEE 47th*, volume 2, pages 1123–1127 vol.2.
- [68] X. Li. *Efficient Ray Tracing Simulation*. Master’s thesis, Lund University and Forschungszentrum Telekommunikation Wien GmbH, 2014.
- [69] Y. Ding Y. Eghbali H. Muhaidat S. H. Li, L. Ibdah and X. Ma. Indoor multi-wall path loss model at 1.93 GHz. In *Military Communications Conference, MILCOM 2013 - 2013 IEEE*, pages 1233–1237.
- [70] G. Liang and H. L. Bertoni. A new approach to 3D ray tracing for site specific propagation modeling. In *Vehicular Technology Conference, 1997, IEEE 47th*, volume 2, pages 1113–1117 vol.2.
- [71] G. Liang and H. L. Bertoni. A new approach to 3-D ray tracing for propagation prediction in cities. *IEEE Transactions on Antennas and Propagation*, 46(6):853–863, 1998.
- [72] J. Lin and Y. Li. Finding structural similarity in time series data using BOP. *International Conference on Scientific and Statistical Database Management*, pages 461–477, 2009.
- [73] J. Lin, S. Williamson, K. Borne, and D. DeBarr. Pattern recognition in time series. *Advances in Machine Learning and Data Mining for Astronomy*, pages 617–645, 2010.

- [74] Z. Liu, L. Guo, X. Meng, and Z. Zhong. A novel 3D ray-tracing model for propagation prediction in indoor environments. In *Antennas, Propagation and EM Theory (ISAPE), 2012 10th International Symposium on*, pages 428–431.
- [75] Z. Y. Liu, L. X. Guo, and C. L. Li. Effects of antenna polarization on power and RMS delay spread in LOS/OOS indoor radio channel. In *General Assembly and Scientific Symposium (URSI GASS), 2014 XXXIth URSI*, pages 1–4.
- [76] D. C. Livingston. *The physics of microwave propagation*. Prentice-Hall electrical engineering series: Microwaves and fields series. Prentice-Hall, 1970.
- [77] A. Lodhi. Prediction study of LTE received signal strengths at short distances from base stations. Delivery report, SIRADEL Ltd and OFCOM Ltd, 2013.
- [78] H. Lohrasbi-peydeh, T. A. Gulliver, and H. Amindavar. Unknown transmit power RSSD based source localization with sensor position uncertainty. *IEEE Transactions on Communications*, 63(5):1784–1797, 2015.
- [79] K. W. K. Lui and H. C. So. Range-based source localisation with pure reflector in presence of multipath propagation. *Electronics Letters*, 46(13):957–958, 2010.
- [80] L. R. Maciel, H. L. Bertoni, and H. N. Xia. Unified approach to prediction of propagation over buildings for all ranges of base station antenna height. *IEEE Transactions on Vehicular Technology*, 42(1):41–45, 1993.
- [81] B. Malila, O. Falowo, and N. Ventura. Millimeter wave small cell backhaul: An analysis of diffraction loss in NLOS links in urban canyons. In *AFRICON, 2015*, pages 1–5.
- [82] S. Matsuo, M. Gocho, T. Hashimoto, and A. Ozaki. GPU-based parallel algorithm for VPL-approximated EM wave propagation. In *EuCAP 2017*.
- [83] P. Meissner, M. Gan, F. Mani, E. Leitinger, M. Frohle, C. Oestges, T. Zemen, and K. Witrisal. On the use of ray tracing for performance prediction of UWB indoor localization systems. In *Communications Workshops (ICC), 2013 IEEE International Conference on*, pages 68–73.
- [84] B. Meng, W. Shan, H. Li, and F. Li. Construction and implement of semi-physical electromagnetic propagation simulation platform. In *Electronics, Communications and Control (ICECC), 2011 International Conference on*, pages 783–787.
- [85] Y. Miao, Q. Gueuning, M. Gan, and C. Oestges. Adding diffuse scattering correlation to effective roughness models in ray tracing. In *EuCAP 2017*.

- [86] Harold Mott. *Polarization in Antennas and Radar*. Wiley-Interscience publication, 1986.
- [87] J. Nuckelt, T. Abbas, F. Tufvesson, C. Mecklenbrauker, L. Bernado, and T. Kurner. Comparison of ray tracing and channel-sounder measurements for vehicular communications. In *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pages 1–5.
- [88] J. Nuckelt, D. M. Rose, T. Jansen, and T. Kurner. On the use of openstreetmap data for V2X channel modeling in urban scenarios. In *Antennas and Propagation (EuCAP), 2013 7th European Conference on*, pages 3984–3988.
- [89] P. Okvist, H. Asplund, A. Simonsson, B. Halvarsson, J. Medbo, and N. Seifi. 15 GHz propagation properties assessed with 5G radio access prototype. In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2015 IEEE 26th Annual International Symposium on*, pages 2220–2224.
- [90] D. Pena, R. Feick, H. D. Hristov, and W. Grote. Measurement and modeling of propagation losses in brick and concrete walls for the 900-MHz band. *Antennas and Propagation, IEEE Transactions on*, 51(1):31–39, 2003.
- [91] H. Peter. *Machine Learning In Action*. Manning Publications Co., Shelter Island, NY 11964, 2012.
- [92] B. R. Phelan, E. H. Lenzing, and R. M. Narayanan. Source localization using unique characterizations of multipath propagation in an urban environment. In *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2012 IEEE 7th*, pages 189–192.
- [93] Y. Pinhasi, A. Yahalom, and S. Petnev. Propagation of ultra wide-band signals in lossy dispersive media. In *Microwaves, Communications, Antennas and Electronic Systems, 2008. COMCAS 2008. IEEE International Conference on*, pages 1–10.
- [94] L. A. R. Ramirez, L. A. R. Silva Mello, and F. J. V. Hasselmannn. Application of a ray tracing technique for coverage predictions in micro cellular environments. In *Microwave and Optoelectronics Conference, 2003. IMOC 2003. Proceedings of the 2003 SBMO/IEEE MTT-S International*, volume 1, pages 171–175 vol.1.
- [95] J. Richter, M. O. Al-Nuaimi, and L. P. Ivriissimtzis. Optimisation of radio coverage in urban microcells using a UTD based ray-tracing model. *Microwaves, Antennas and Propagation, IEEE Proceedings*, 151(3):187–192, 2004.

- [96] J. Richter, R. F. S. Caldeirinha, M. O. Al-Nuaimi, A. Seville, N. C. Rogers, and N. Savage. A generic narrowband model for radiowave propagation through vegetation. In *2005 IEEE 61st Vehicular Technology Conference*, volume 1, pages 39–43 Vol. 1.
- [97] K. Rizk, A. Mawira, J. F. Wagen, and F. Gardiol. Propagation in urban micro-cells with high rise buildings. In *Vehicular Technology Conference, 1996. Mobile Technology for the Human Race., IEEE 46th*, volume 2, pages 859–863 vol.2.
- [98] N. C. Rogers. A generic model of 1-60 GHz radio propagation through vegetation. Report, Radiocommunications Agency, 2002.
- [99] J. A. Romo, I. F. Anitzine, and F. P. Fontan. Application of neural networks to field strength prediction for indoor environments. In *Antennas and Propagation, 2006. EuCAP 2006. First European Conference on*, pages 1–6.
- [100] C. Saeidi, F. Hodjatkashani, and A. Fard. New tube-based shooting and bouncing ray tracing method. In *Advanced Technologies for Communications, 2009. ATC '09. International Conference on*, pages 269–273.
- [101] A. A. M. Saleh and R. A. Valenzuela. A statistical model for indoor multipath propagation. *Selected Areas in Communications, IEEE Journal on*, 5(2):128–137, 1987.
- [102] S. Salous. *Radio Propagation Measurement and Channel Modelling*. Wiley, 1st edition, 2013.
- [103] S. Salous, A. Cheema, and X. Raimundo. Radio channel propagation measurements using a multiband agile chirp sounder. In *General Assembly and Scientific Symposium (URSI GASS), 2014 XXXIth URSI*, pages 1–4.
- [104] J. Sanders and K. Edward. *CUDA by Example: an introduction to general-purpose GPU programming*. Pearson Education Inc., 2010.
- [105] S. Saunders. *Antennas and Propagation for Wireless Communication Systems*. John Wiley and Sons Ltd, 2nd edition, 2007.
- [106] K. R. Schaubach, N. J. Iv Davis, and T. S. Rappaport. A ray tracing method for predicting path loss and delay spread in microcellular environments. In *Vehicular Technology Conference, 1992, IEEE 42nd*, pages 932–935 vol.2.
- [107] M. Schiller, A. Knoll, M. Mocker, and T. Eibert. GPU accelerated ray launching for high-fidelity virtual test drives of VANET applications. In *High Performance*

*Computing and Simulation (HPCS), 2015 International Conference on*, pages 262–268.

- [108] R. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, 34(3):276–280, 1986.
- [109] A. Schmitz and L. Kobbelt. Efficient and accurate urban outdoor radio wave propagation. In *Electromagnetics in Advanced Applications (ICEAA), 2011 International Conference on*, pages 323–326.
- [110] J. Schmitz, F. Schröder, and R. Mathar. TDOA fingerprinting for localization in non-line-of-sight and multipath environments. In *2015 International Symposium on Antennas and Propagation (ISAP)*, pages 1–4.
- [111] E. Semaan, F. Harrysson, A. Furuskar, and H. Asplund. Outdoor-to-indoor coverage in high frequency bands. In *Globecom Workshops (GC Wkshps), 2014*, pages 393–398.
- [112] S. A. A. Shahidian and H. Soltanizadeh. Autonomous trajectory control for limited number of aerial platforms in RF source localization. In *Robotics and Mechatronics (ICROM), 2015 3rd RSI International Conference on*, pages 755–760.
- [113] Y. Shu, Y. Huang, J. Zhang, P. Coue, P. Cheng, J. Chen, and K. G. Shin. Gradient-based fingerprinting for indoor localization and tracking. *IEEE Transactions on Industrial Electronics*, 63(4):2424–2433, 2016.
- [114] T. Sofos and P. Constantinou. Propagation model for vegetation effects in terrestrial and satellite mobile systems. *IEEE Transactions on Antennas and Propagation*, 52(7):1917–1920, 2004.
- [115] S. P. Sotiroudis, S. K. Goudos, K. A. Gotsis, K. Siakavara, and J. N. Sahalos. Application of a composite differential evolution algorithm in optimal neural network design for propagation path-loss prediction in mobile communication systems. *Antennas and Wireless Propagation Letters, IEEE*, 12:364–367, 2013.
- [116] Q. Spencer, M. Rice, B. Jeffs, and M. Jensen. A statistical model for angle of arrival in indoor multipath propagation. In *Vehicular Technology Conference, 1997, IEEE 47th*, volume 3, pages 1415–1419 vol.3.
- [117] S. Spiegel, J. Gaebler, A. Lommatzsch, E. D. Luca, and S. Albayrak. Pattern recognition and classification for multivariate time series. *SensorKDD’11*, 2011.

- [118] A. Tayebi, J. Gomez, F. S. de Adana, and O. Gutierrez. Ray-tracing application to mobile localization in multipath indoor environments. In *Electromagnetics in Advanced Applications, 2009. ICEAA '09. International Conference on*, pages 412–415.
- [119] F. Tila, P. R. Shepherd, and S. R. Pennock. Indoor ray tracing evaluation of enhanced high frequency communications using directional antennas at the base-station and space diversity at the terminal. In *Microwave Conference, 2001. 31st European*, pages 1–5.
- [120] S. A. Torrico, K. L. Chee, and T. Kurner. A propagation prediction model in vegetated residential environments - a simplified analytical approach. In *Proceedings of the 5th European Conference on Antennas and Propagation (EuCAP)*, pages 3279–3283.
- [121] C. Tzaras and S. R. Saunders. Rapid, uniform computation of multiple knife-edge diffraction. *Electronics Letters*, 35(3):237–239, 1999.
- [122] F. T. Ulaby, R. K. Moore, and A. K. Fung. *Microwave Remote Sensing Fundamentals and Radiometry*, volume 1 of *Microwave Remote Sensing Active and Passive*. ARTECH HOUSE, 1981.
- [123] R. L. Vacarro and J. J. Fante. Cancellation of jammers and jammer multipath in a GPS. *IEEE Aerospace and Electronic Systems Magazine*, pages 25 – 28, 1998.
- [124] N. Sipus Z. Vilovic, I. Burum. A comparison of neural network models for indoor field strength prediction. In *ELMAR, 2007*, pages 251–254.
- [125] W. J. Vogel and G. W. Torrence. Propagation measurements for satellite radio reception inside buildings. *Antennas and Propagation, IEEE Transactions on*, 41(7):954–961, 1993.
- [126] A. Wadhwa, U. Madhow, J. Hespanha, and B. M. Sadler. Following an RF trail to its source. In *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 580–587.
- [127] B. Wagner, D. Timmermann, G. Ruscher, and T. Kirste. Device-free user localization utilizing artificial neural networks and passive RFID. In *2012 Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS)*, pages 1–7.

- [128] D. J. Walter, K. Bryan, J. Stephens, C. Bullmaster, and V. Chakravarthy. Localization of RF emitters using compressed sensing with multiple cooperative sensors. In *2012 IEEE National Aerospace and Electronics Conference (NAECON)*, pages 236–240.
- [129] P. Wawrzyniak, S. Hausman, and P. Korbelt. Area based indoor tracking algorithm based on sequence detection and maximum likelihood metrics. In *2016 10th European Conference on Antennas and Propagation (EuCAP)*, pages 1–3.
- [130] P. Wawrzyniak, S. Hausman, and P. Korbelt. Sequence detection of movement for accurate area based indoor positioning and tracking. In *2015 9th European Conference on Antennas and Propagation (EuCAP)*, pages 1–4.
- [131] J. Weng, X. Tu, Z. Lai, S. Salous, and J. Zhang. Modelling the mm-wave channel based on intelligent ray launching model. In *Antennas and Propagation (EuCAP), 2015 9th European Conference on*, pages 1–4.
- [132] R. Wilfinger, T. Moder, M. Wieser, and B. Grosswindhager. Indoor position determination using location fingerprinting and vehicle sensor data. In *2016 European Navigation Conference (ENC)*, pages 1–9.
- [133] X. Wu, V. Kumar, J. Ross, et al. Top 10 algorithms in data mining. In *IEEE International Conference on Data Mining (ICDM)*, 2007.
- [134] H. H. Xia. An analytical model for predicting path loss in urban and suburban environments. In *Personal, Indoor and Mobile Radio Communications, 1996. PIMRC'96., Seventh IEEE International Symposium on*, volume 1, pages 19–23 vol.1.
- [135] C. Yang, B. Wu, and C. Ko. A ray-tracing method for modeling indoor wave propagation and penetration. *Antennas and Propagation, IEEE Transactions on*, 46(6):907–919, 1998.
- [136] G. Yang, K. Pahlavan, and J. F. Lee. A 3D propagation model with polarization characteristics in indoor radio channels. In *Global Telecommunications Conference, 1993, including a Communications Theory Mini-Conference. Technical Program Conference Record, IEEE in Houston. GLOBECOM '93., IEEE*, pages 1252–1256 vol.2.
- [137] G. Yang, K. Pahlavan, J. F. Lee, A. J. Dagen, and J. Vancraeynest. Prediction of radio wave propagation in four blocks of New York city using 3D ray tracing. In *Personal, Indoor and Mobile Radio Communications, 1994. Wireless Networks -*



*Catching the Mobile Future.*, 5th IEEE International Symposium on, volume 1, pages 263–267 vol.1.

- [138] S. Yun-di Lien and M. Cherniakov. Analytical approach for multipath delay spread power distribution. In *Global Telecommunications Conference, 1998. GLOBECOM 1998. The Bridge to Global Integration. IEEE*, volume 6, pages 3680–3685 vol.6.
- [139] H. Zhao, G. Zheng, L. Zhang, W. Fu, H. Wei, and H. Shao. Ray tracing analysis of Gaussian beamed millimetre wave propagation in circle tunnel. In *Wireless Mobile and Computing (CCWMC 2009), IET International Communication Conference on*, pages 164–167.
- [140] Y. Zhao and Y. Zhuo. The algorithm of position related channel character extraction and match localization based on ray-tracing. In *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, volume 2, pages V2–429–V2–433.
- [141] H. Zou, B. Huang, X. Lu, H. Jiang, and L. Xie. A robust indoor positioning system based on the procrustes analysis and weighted extreme learning machine. *IEEE Transactions on Wireless Communications*, 15(2):1252–1266, 2016.

# Appendix

## A. Overpass Turbo map geometry extraction function based on Matlab R2016b

Function S = MapExtract3( ) % Convert Overpass Turbo coordinates into Matlab structure files

Place0 = kml2struct('Hoxton1.kml'); % Read structural data of the Hoxton scenario from file

Os = [-0.093430, 51.528708]; % The origin for Hoxton scenario

rmax = 6378137; % Maximum radius of the earth

rmin = 6356752.3142; % Minimum radius of the earth

r = sqrt(((rmax<sup>2</sup> · cos(Os(2)))<sup>2</sup> + (rmin<sup>2</sup> · sin(Os(2)))<sup>2</sup>)/

((rmax · cos(Os(2)))<sup>2</sup> + (rmin · sin(Os(2)))<sup>2</sup>));

i=1;

for t=1:size(Place0,2)

if strcmp(Bath0(t).Name, 'undefined')==0

Place1,i = [Place0(t).Lon Place0(t).Lat];

i=i+1;

end;

end;

Place\_S = cell(1,i-1);

for q=1:size(Place,2)

P = size(Place{1,q},1);

for p=1:P

if isnan(Place{1,q}(p,:)) =[0 0]

break;

else

Place\_S{1,q}(p,:) = [sind(Place{1,q}(p,1)-Os(1))·r, sind(Place{1,q}(p,2)-Os(2))·r];

end;

end;

end;

S = [ ];

for q=1:size(Place\_S,2)

```

P = size(Place_S{1,q},1);
for p=1:(P-1)
    S(size(S,1)+1,:) = Places {1,q }(p,:);
    S(size(S,1)+1,:) = Places {1,q }(p+1,:);
end;
end;
m = length(S)/2;
figure(33);
hold on;
grid on;
for k=1:m
    a = min(S(2*k-1,1),S(2*k,1));
    b = max(S(2*k-1,1),S(2*k,1));
    if (a == b)
        x = a:0.01:b;
        y = ((S(2*k,2)-S(2*k-1,2))/(S(2*k,1)-S(2*k-1,1)))*(x-S(2*k-1,1))+S(2*k-1,2);
        plot(x,y,'k-','linewidth',2);
    else
        a0 = min(S(2*k-1,2),S(2*k,2));
        b0 = max(S(2*k-1,2),S(2*k,2));
        y = a0:0.01:b0;
        plot(a*ones(size(y)),y,'k-','linewidth',2);
    end;
end;
axis equal;

end

```

## B. Ray-surface intersection detection based on Visual Studio Community 2015 using CUDA8.0

```
# include "stdafx.h"
# include <stdio.h>
# include <stdlib.h>
# include <iostream>
# include <math.h>
# include <device_launch_parameters.h>
# include <cuda_runtime.h>
# include <cuda.h>
# include "mex.h"
# define n 2
# define S 5448
# define Pi 3.1415926535897932384626

void intersect0(float *dev_a , float *dev_b , float *dev_t , int *dev_flag, float *Ori,
float *Dir);

void mexFunction(int nlhs, mxArray *plhs[ ], int nrhs, const mxArray *prhs[ ]) {

    nrhs = 4;
    double *valueO = mxGetPr(prhs[0]);
    double *valueD = mxGetPr(prhs[1]);
    float origin[n], direction[n];
    origin[0] = (float)valueO[0];
    origin[1] = (float)valueO[1];
    direction[0] = (float)valueD[0];
    direction[1] = (float)valueD[1];
    mwSize row, col;
    mwSize nrow = mxGetM(prhs[2]);
    mwSize ncol = mxGetN(prhs[2]);
    double *values = mxGetPr(prhs[2]);
    double *valuesB = mxGetPr(prhs[3]);
    cudaFree(0);
    int *dev_flag, flag[S];
    float *dev_t, t[S], *dev_a, *dev_b;
```

```

float a[S][n];
float b[S][n];
cudaHostAlloc((void**)& a, n*S * sizeof(float), cudaHostAllocDefault);
cudaHostAlloc((void**)& b, n*S * sizeof(float), cudaHostAllocDefault);
for (row = 0; row < nrow; row++) {
for (col = 0; col < ncol; col++) {
a[row][col] = (float)values[nrow * col + row];
b[row][col] = (float)valuesB[nrow * col + row];
}
}
cudaHostAlloc((void**)& flag, S * sizeof(int), cudaHostAllocDefault);
cudaHostAlloc((void**)& t, S * sizeof(float), cudaHostAllocDefault);
cudaMalloc((void**)& dev_ flag, S * sizeof(int));
cudaMalloc((void**)& dev_ t, S * sizeof(float));
cudaMalloc((void**)& dev_ a, n*S * sizeof(float));
cudaMalloc((void**)& dev_ b, n*S * sizeof(float));
cudaMemcpyAsync(dev_ a, a, n*S * sizeof(float), cudaMemcpyHostToDevice);
cudaMemcpyAsync(dev_ b, b, n*S * sizeof(float), cudaMemcpyHostToDevice);

```

```

intersect0(dev_ a, dev_ b, dev_ t, dev_ flag, origin, direction);

```

```

cudaMemcpyAsync(flag, dev_ flag, S * sizeof(int), cudaMemcpyDeviceToHost);
cudaMemcpyAsync(t, dev_ t, S * sizeof(float), cudaMemcpyDeviceToHost);
nlhs = 2;
double mexFlag[S], mexT[S];
for (int j = 0; j < S; j++) {
mexFlag[j] = (double)flag[j];
mexT[j] = (double)t[j];
}
plhs[0] = mxCreateDoubleMatrix(1, S, mxREAL);
plhs[1] = mxCreateDoubleMatrix(1, S, mxREAL);
memcpy(mxGetPr(plhs[0]), mexFlag, S * sizeof(double));
memcpy(mxGetPr(plhs[1]), mexT, S * sizeof(double));
cudaFreeHost(a);
cudaFreeHost(b);
cudaFree(dev_ a);

```

```

    cudaFree(dev_ b);
    cudaFreeHost(t);
    cudaFreeHost(flag);
    cudaFree(dev_ t);
    cudaFree(dev_ flag);
    return;
}

```

```

__ constant__ float o[n];
__ constant__ float d[n];

```

```

__ global__ void intersectMEX(float *a, float *b, float *t, int *flag) {

    int id = blockIdx.x * blockDim.x + threadIdx.x;
    if (id >= S) {
        return;
    }
    if (id < S) {
        float line[n] = { a[id*n + 0] - b[id*n + 0], a[id*n + 1] - b[id*n + 1] };
        float l[n] = { line[0] / sqrt(line[0] * line[0] + line[1] * line[1]), line[1] / sqrt(line[0] *
line[0] + line[1] * line[1]) };
        float ParallelCheck = -l[1] * d[0] + l[0] * d[1];
        float x0;
        if (d[0] < 0) {
            x0 = -d[0];
        }
        else {
            x0 = d[0];
        }
        float y0;
        if (d[1] < 0) {

```

```

y0 = -d[1];
}
else {
y0 = d[1];
}
if (ParallelCheck <= 0.0001 && ParallelCheck >= -0.0001) {
float ao[ ] = { a[id*n + 0] - o[0], a[id*n + 1] - o[1] };
float bo[ ] = { b[id*n + 0] - o[0], b[id*n + 1] - o[1] };
float OverlapCheck = -ao[1] * bo[0] + ao[0] * bo[1];
if (OverlapCheck <= 0.0001 && OverlapCheck >= -0.0001) {
if (x0 < 0.01) {
float ta = a[id*n + 1] - o[1];
if (ta < 0) { ta = -ta; }
float tb = b[id*n + 1] - o[1];
if (tb < 0) { tb = -tb; }
float tAB[2][n] = { ao[0],ao[1],bo[0],bo[1] };
float tt[2] = { ta, tb };
if ((tt[0] + tt[1]) > sqrt(line[0] * line[0] + line[1] * line[1])) {
int index;
if (tt[0] < tt[1]) {
t[id] = tt[0];
index = 0;
}
else {
t[id] = tt[1];
index = 1;
}
if ((tAB[index][0] * d[0] + tAB[index][1] * d[1]) > 0) {
flag[id] = -1;
return;
}
else {
flag[id] = 0;
t[id] = 0;
return;
}
}
}
}

```

```

else {
flag[id] = 0;
t[id] = 0;
return;
}
}
if (y0 < 0.01) {
float ta = a[id*n + 0] - o[0];
if (ta < 0) { ta = -ta; }
float tb = b[id*n + 0] - o[0];
if (tb < 0) { tb = -tb; }
float tAB[2][n] = { ao[0],ao[1],bo[0],bo[1] };
float tt[2] = { ta, tb };
if ((tt[0] + tt[1]) > sqrt(line[0] * line[0] + line[1] * line[1])) {
int index;
if (tt[0] < tt[1]) {
t[id] = tt[0];
index = 0;
}
else {
t[id] = tt[1];
index = 1;
}
if ((tAB[index][0] * d[0] + tAB[index][1] * d[1]) > 0) {
flag[id] = -1;
return;
}
else {
flag[id] = 0;
t[id] = 0;
return;
}
}
else {
flag[id] = 0;
t[id] = 0;
return;
}
}

```



```

}
}
float ta = (a[id*n + 1] - o[1]) / d[1];
if (ta < 0) { ta = -ta; }
float tb = (b[id*n + 1] - o[1]) / d[1];
if (tb < 0) { tb = -tb; }
float tAB[2][n] = { ao[0],ao[1],bo[0],bo[1] };
float tt[2] = { ta, tb };
if ((tt[0] + tt[1]) > (0.1 + sqrt(line[0] * line[0] + line[1] * line[1]))) {
int index;
if (tt[0] < tt[1]) {
t[id] = tt[0];
index = 0;
}
else {
t[id] = tt[1];
index = 1;
}
if ((tAB[index][0] * d[0] + tAB[index][1] * d[1]) > 0) {
flag[id] = -1;
return;
}
else {
flag[id] = 0;
t[id] = 0;
return;
}
}
}
flag[id] = 0;
t[id] = 0;
return;
}
float k;
if (d[0]<0.01 && d[0]>-0.01) {
k = (o[0] - a[id*n + 0]) / (b[id*n + 0] - a[id*n + 0]);
t[id] = (a[id*n + 1] - o[1] + k*(b[id*n + 1] - a[id*n + 1])) / d[1];

```

```

    }
    else {
        float nomi = (a[id*n + 1] - o[1])*d[0] - (a[id*n + 0] - o[0])*d[1];
        float deno = (b[id*n + 0] - a[id*n + 0])*d[1] - (b[id*n + 1] - a[id*n + 1])*d[0];
        k = nomi / deno;
        t[id] = ((a[id*n + 0] - o[0]) + k*(b[id*n + 0] - a[id*n + 0])) / d[0];
    }
    if (t[id] < 0.1) {
        flag[id] = 0;
        t[id] = 0;
        return;
    }
    else if (k <= -0.001 || k >= 1.001) {
        flag[id] = 0;
        t[id] = 0;
        return;
    }
    else {
        if (k > -0.001 && k < 0.001) {
            flag[id] = -2;
            return;
        }
        else if (k > 0.999 && k < 1.001) {
            flag[id] = -2;
            return;
        }
        else {
            flag[id] = 1;
            if (d[0]<0.01 && d[0]>-0.01) { t[id] - = 0.1; }
        }
    }
}

```

```

void intersect0(float *dev_a, float *dev_b, float *dev_t, int *dev_flag, float *origin,
float *direction) {

    cudaMemcpyToSymbol(o, origin, n * sizeof(float));
    cudaMemcpyToSymbol(d, direction, n * sizeof(float));
    dim3 blocksPerGrid((S + 511) / 512, 1);
    dim3 threadsPerBlock(512, 1)

    intersectMEX <<< blocksPerGrid, threadsPerBlock >>> (dev_a, dev_b, dev_t,
dev_flag);

}

```

## C. List of publications

[1]. Z. Dai and R. J. Watson, “A GPU accelerated ray-launching model using CUDA,” 2018 12th European Conference on Antennas and Propagation (EuCAP), London, 2018.

[2]. Z. Dai, P. R. Shepherd and R. J. Watson, “UAV-aided source localization in urban environments based on ray launching simulation,” 2017 IEEE AP-S Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting, San Diego, USA, 2017.

[3]. Z. Dai, R. J. Watson and P. R. Shepherd, “A propagation modeling approach to navigation,” 2017 11th European Conference on Antennas and Propagation (EuCAP), Paris, 2017.

[4]. Z. Dai, R. J. Watson and P. R. Shepherd, “A propagation modeling approach to source location and navigation,” 2016 10th European Conference on Antennas and Propagation (EuCAP), Davos, 2016.